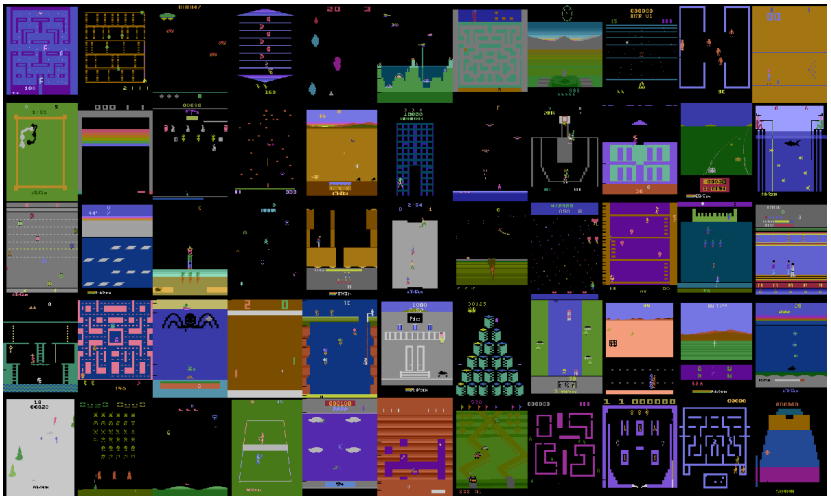
Is Deep Reinforcement learning Superhuman on Atari?

A detailed analysis of Deep Reinforcement learning agents performing in the Arcade learning environment



Lisa Golla
lgolla@uni-osnabrueck.de

UNIVERSITÄT OSNABRÜCK

A Term paper for the course Deep reinforcement learning
Prof. Dr. Elia Bruni, Prof. Dr. Gordon Pipa
SS 2022

Cognitive science
University Osnabrück
Germany, Osnabrück
September 2022

Is Deep Reinforcement learning Superhuman on Atari?

—The *Arcade learning environment (ALE)* is offering 57 Atari games which are used in the Deep Reinforcement learning (DRL) community as a benchmark to evaluate the generality of a DRL agent. There have been several improvements on agents and also a lot of different approaches. Giving an overview, seven different DRL agents and their performance on the Atari benchmark are shown. In order to make a proper comparison possible, the most important metrics and also the current state of affairs is presented regarding challenges, achievements and goals. During the course of this paper the meaning for DRL will be highlighted. Connected to that the question is raised whether first the current state of the art agent GDI-I^3 is superhuman and second whether DRL is generally superhuman on Atari. Additionally, the underlying *Generalized Data Iteration (GDI)* algorithm will be explained in more detail for the purpose of showing the current state of affairs in research since GDI-I^3 is the current state of the art (SOTA) agent which has outperformed all other approaches so far in the mean HNS, HWRB and HWRNS metrics..

***Index Terms*—Deep Reinforcement learning, Atari, Arcade learning environment, Generalized data distribution Iteration**

I. Introduction

For years already the field Artificial Intelligence (AI) is specifically interested in developing algorithms which are capable of general competency [1], [2], [4]. The idea is that it is possible to apply one algorithm in various tasks as well as domains and therefore remove the need for domain specific tailoring. While considering the idea of a general competency, a question is raised: what is the best method to evaluate agents? Or, in other words, how can it be decided whether an agent has gained general competency or not? There is a new confrontation with finding suitable measurements and benchmarks for the purpose of evaluation [1].

During the course of this paper the Arcade learning environment (ALE) is introduced. ALE is providing an interface to over hundred reproduced Atari 2600 game environments. Atari 2600, a second generation game console, was originally released in 1977 and remained massively popular for over a decade. Over 500 games were developed for the Atari 2600, spanning a diverse range of genres such as shooters, beat'em ups, puzzle, sports, and action-adventure games; many game genres were pioneered on the console [4]. While modern game consoles involve visuals, controls, and a general complexity that rivals the real world, Atari 2600 games are far simpler. Each of the games is created in a different, as well as interesting way so that the process of solving the games is a challenge for human players. Generally, ALE is associated with several research challenges with regard to reinforcement learning, model learning. model-based planning and also transfer learning [1], [3], [4]. In the context of this paper the main feature to be highlighted is providing a suitable test environment which enables evaluation and comparing approaches for the named challenges. Why is ALE interesting for Machine learning (ML) applications? Usually when training a ML algorithm on the same data set it is considered to only do poor experimental practice when it comes to training and evaluation since there is the well-known risk of overestimating the algorithm's performance. Therefore, it is a common practice to do training on one dataset and afterwards do the evaluation on a disjoint test set [4]. The high amount of available games in ALE make a similar procedure possible. In a sense, the domain representation can be first tuned on a small number of training games and afterwards tested on new, unseen testing games. While general competency remains the long-term goal for artificial intelligence, ALE proposes an achievable stepping stone: techniques for general competency across the spectrum of Atari 2600 games. In other words, if there is a ML algorithm that can solve given Atari 2600 games successfully, or to a satisfactory degree, this means that the research is one step closer to the goal of gaining general competency for given agents.

In the DRL community Atari games have been a long-standing benchmark for the last decade [2], [6], [9]. As indicated above, they used this benchmark to test the general competency of DRL algorithms.

Historically speaking, there are a lot of different approaches and algorithms to solve the Atari games of the ALE. The Rainbow DQN [6] is well known to be a state of the art DQN agent which fruitfully combined six extensions of the DQN family. The DQN family was one of the first algorithms performing pretty well on Atari. Also here were developed several improvements and revisions [10], for example in terms of interpretability the so-called Region-sensitive Rainbow (RS-Rainbow) was developed [11], or in terms of data efficiency and velocity there were some improvements done on Rainbow DQN [13], as well as some improvements based on the guarantee of robustness in presence of large noise by adding a proximal term, called Rainbow Pro [12]. It is worth noting that still until 2022 there are some improvements made for Rainbow DQN highlighting its importance in research. Next to the model-free approach, simultaneously there were also some model-based algorithms developed like SIMPLE [8] and MuZero [14]. Moreover, there were some stepping stones reached like overcoming the high exploration problem as Go-Explore achieved it [7], or achieving the human benchmark in Atari as Agent57 reached it in all 57 games, and also GDI-

H^3 which is known for superhuman performance in terms of breaking several human world records [3].

The paper at hand is meant to be a summary paper. The main contributions are

1) providing and explaining the background knowledge of the Atari human benchmark containing metrics, achievements, goals and challenges
2)summarizing the GDI algorithm and connect the performance of the GDI-H^3 agent to other agents as well as explaining what the algorithm solved and how,
3) Answering the question whether DRL is superhuman on Atari, name some critical thoughts and raise some further questions for future research.

II. SOTA ALGORITHMS IN ATARI

A. Model-based and model-free approaches

RL algorithms can be mainly divided into two categories – model-based and model-free.On the one hand, Model-based RL has an agent aiming to understand its environment and build a model for it based on its interactions with this environment. In such a system, preferences take priority over the consequences of the actions. As a consequence, the greedy agent will always try to perform an action that will get the maximum reward irrespective of what that action may cause [8].

On the other hand, model-free RL algorithms try to learn the consequences of their actions through experience. Algorithms such as for example Policy Gradient, or Q-Learning, are used for that. In a nutshell, such algorithms will need an action multiple times and will adjust the policy, or strategy, for optimal rewards, based on the outcomes [8]. Further, if the agent can predict the reward for some action before performing it, its actions are planned and therefore the algorithm is classified as model-based. Whereas, if the agent actually has to do the action to see what happens and learn from it, it is model-free.

Due to their different approaches, those two branches of RL algorithms are used for different applications. For instance, model-based approaches are more suitable for full information games like chess. Generally speaking the applications are more in fields where the environment is static and efficiency is the main concern. When it comes to real world applications, like self-driving cars, it would be more useful to use a model-free approach since it can adapt better to an environment by learning through actions than being pre-defined, static. Also creating a proper model would be quite time-consuming in complex real-world situations. [1], [9]

B. A representative selection

In the Atari game context both, model-free and model-based approaches are used. In order to give a representative, multifaceted view of the current Atari benchmark in DRL the performance of seven DRL agents will be compared along the course of this paper which claimed to reach SOTA performance.

- **Rainbow DQN**
  Rainbow [6] is a classic value based RL algorithm and part of the DQN algorithm family. It has successfully combined six extensions of the DQN algorithm family. Historically speaking it is recognized to achieve state-of-the-art performance on the ALE benchmark.
- **SIMPLE**
  SimPLe [8], or Simulated Policy Learning is one of the classic model-based RL algorithms on Atari. It adopted a video prediction model to enable RL agents to solve Atari problems with higher sample efficiency. It claimed to outperform the SOTA model-free algorithms in most games.
- **Go-Explore**
  Go-Explore [7] has achieved SOTA in Atari and overcame the hard exploration problem. Go-Explore adopted three principles to solve this problem. Firstly, agents remember previously visited states. Secondly, agents first return to a promising state and then explore it. Finally, solve the simulated environment through any available means, and then robustify via imitation learning.
- **Agent57**
  Agent57 [5] is the current SOTA model-free RL algorithms on Median Human normalized score of the Atari Benchmark. Agent57 proposed a novel state-action value function parameterization method and adopted an adaptive exploration over a family of policies. It is known as the first one achieving superhuman performance on all 57 Atari games of the ALE environment.
- **MuZero**
  Muzero [14] has combined a tree-based search with a learned model. It is also a model-based algorithm and has proven to achieve superhuman performance.
- **GDI-I^3 & GDI-H^3**
  GDI [3], or Generalized Data Distribution Iteration, claimed to have achieved SOTA on mean/median Human world record normalized score, mean Human normalized score, and Human world record breakthrough of Atari Benchmark. Therefore, it is known as the current SOTA algorithm of the Atari benchmark. GDI is one of the novel Reinforcement Learning paradigms, which combined a data distribution optimization operator into the traditional generalized policy iteration (GPI) and thus achieved human-level learning efficiency.

The listed agents were chosen, because the selection is pretty representative. model-free as well as model-based algorithms are included, one of the first algorithms performing SOTA to show the historical progress of the algorithms in performance as well as the current SOTA algorithm in achieving human world records which is known for top performance [3], can well be compared to the second model-free algorithm since it is known for average performance reaching the human benchmark [5]. Go-Explore can be seen as a stepping stone in between since it manages to solve the hard exploration problem which is considered as one of the hardest challenges

to face in the Atari benchmark.

In the past SOTA algorithms were fragile facing the hard exploration problem [1], DQN for example suffered from that. The "hard-exploration" problem refers to exploration in an environment with very sparse or even deceptive reward. It is difficult because random exploration in such scenarios can rarely discover successful states or obtain meaningful feedback. Montezuma's Revenge is a concrete example for the hard-exploration problem. It is known as a challenging game in Atari for DRL to solve [1], [2]. Agent57, GoExplore and MuZero have overcome this problem, but failed to balance the trade-off between exploration and exploitation leading to lower learning efficiency. Learning from sparse rewards is extremely difficult for model-free RL algorithms [1], [5], especially for those without intrinsic rewards that struggle to learn from weak gradient signals. Model-based methods like SIMPLE can ease this problem by adopting a world model for planning or replay [9], which both enhance the gradient signals. However, being utterly dependent on planning is unrealistic. Additionally, such algorithms are losing generality. Model-free RL methods typically obtain remarkable final performance via finding a way to encourage exploration and improve the data richness that guarantees traversal of all possible conditions which is leading to a low sample efficiency [1]. It was a well known problem of combining high sample efficiency and top performance simultaneously whereas GDI can potentially serve as a solution to all of these problems. As will be elaborated in section III, new SOTA algorithms came with new challenges and also the need for new, or different evaluation metrics for the performance. Since GDI is known to be the current SOTA on the current recommended evaluation metrics Human world record breakthrough, this opens up the discussion for section VI. where the question is raised if DRL is actually superhuman in the Atari benchmark since it reached the threshold of the current evaluation metrics.

III. Suitable evaluation systems

Connected to the question whether RL algorithms perform superhumanly, it is important to define an evaluation system which serves as a suitable benchmark. This makes it possible to compare different algorithms with each other. Historically, the given metrics were especially used in order to decide which of the current algorithms can be called the current state of the art algorithm on the basis of generality and performance in a specific metrics. As it will be shown in the next paragraphs the recommended metrics changed over time, because the focus was laid on different skills the algorithms should have, like for example learning efficiency or the breaking of human world records [1]. It will be discussed how important a metric is for the issue of comparing algorithms regarding their performance and how different metrics changed the notion of what the current SOTA algorithm is. Finally, it will be emphasized what the current recommended metric is and how it could be improved using the metrics in future. In order to represent the most used metrics in current research and to give a useful overview for the current topic, it was chosen to introduce the so-called HNS, HWRNS, HWRB, and learning efficiency metrics as well as their (dis)advantages which will be explained further during the course of this section.

A. Human Normalized Score (HNS)

As it was already discussed, reinforcement learning aims at agents that are achieving a superhuman performance. As a consequence, there is a need for a metric that is able to reflect the performance of an algorithm in comparison to a human performance. The calculation can be seen in the following.

$$HNS_{g,i} = \frac{G_{g,i} - G_{g,random}}{G_{g,human\ \ average} - G_{g,random}}$$

In the formula, g is the g_{th} game of Atari, i indicates the algorithm, G_g, human average denotes the human average score baseline and G_g, random stands for the performance of a random policy. The human average score has the advantage that it offers a quite intuitive comparison with human performance. $HNS_{g,i} \geq 100\ \%$ simply means that the algorithm i has reached the human average performance in game g. Consequently, the measure is directly reflecting whether a RL agent reached the average human performance in a specific game. Moreover, the human normalized score makes it possible to easily compare two algorithms with each other since the value $HNS_{g,i}$ is denoting the degree to which a certain algorithm i is surpassing the average level of humans in game g. Therefore, it is possible to contrast one consistent value that expresses the performance of an agent compared to the average human performance [1].

Based on the Human normalized score, the mean HNS is used to represent the mean performance across all 57 Atari games. By contrast, the median HNS was declared to be more reasonable when comparing algorithms due to the fact that as against the mean HNS the median HNS is not susceptible to interference from individual high-scoring games. The question is, is the HNS metrics convenient? The answer is no since the metric is having two problems. Namely, the median HNS is well representing the mediocre performance of an agent, but lacking the information on the top performance. One algorithm can easily achieve a high median HNS, and at the same time a poor mean HNS by adjusting the hyperparameters of algorithms for games near the median score [1]. These metrics can show the generality of the algorithms but fail to reflect the general potential that an algorithm may exhibit. Furthermore, it is important to note here that such metrics would therefore also lead the research to aim at mediocre methods. Generally speaking the metrics can be used well to show the final performance or generality of an algorithm. If the mean or median HNS is more representative in terms of generality or performance is still an open discussion question [1] and its discussion lasts for years already. What can be said is that both metrics are serving different purposes and it may be useful in order to gain as much information as possible to use both of these to evaluate an algorithm.

B. Human world record normalized score (HWRNS)

As it was discussed in section III-A using the HNS may hinder the research to reach the goal of a superhuman agent since there is the tendency to aim at mediocre methods. What can be inferred from this train of thought is that there is a need for metrics that better represent the top performance of an algorithm. To realize that, the HWRNS is depicting the performance compared to the human world record.

$$HWRNS_{g,i} = \frac{G_{g,i} - G_{g,random}}{G_{g,human\ \ world\ \ records} - G_{g,random}}$$

Here, g means the g_{th} game of Atari, i is denoting the RL algorithm, $G_{i,\ human}$ represents the human world records and $G_{g,\ random}$ is the performance of a random policy. Similar to the HNS, HWRNS is an intuitive evaluation metric. It is possible to directly reflect in which games the RL agents surpass the human world records. Additionally, it is also possible to easily compare two algorithms with each other since the value of $HWRNS_{g,i}$ is representing the degree to which algorithm i has surpassed the human world records in game g which is easily comparable. According to that, the mean and median HWRNS represent the mean and median performance of the algorithms across the 57 Atari games. Compared to the mean and median HNS, the mean and median HWRNS have higher requirements for the algorithm to fulfill. It requires the algorithms to pursue a better performance across all the games rather than focus on only a few games [1].

C. Human World Record Breakthrough (HWRB)

It is necessary to put higher requirements on the algorithms in order to prove RL agents are achieving a real superhuman performance. The HWRB can potentially serve as the metric to reveal whether the algorithm has achieved real superhuman performance [1]. It can be calculated as shown below. As can be seen, it simply denotes the fact how many of the world records were achieved by a specific algorithm.

$$HWRB = \sum_{i=1}^{57} (HWRNS \geq 1)$$

D. Learning efficiency

Traditional SOTA algorithms usually are not facing the low learning efficiency problem which is dealing with the fact that data that is used for training is continuously growing [1]. What goes with increasing the training volume is the issue of preventing the application of RL algorithms into the real world. In order to integrate the learning efficiency when comparing to algorithms there is one metric that can be used, namely the game time. Game time is a special feature in Atari and denotes the real time gameplay [1]. It can be calculated as shown below. For example, 200M training frames would be equal to 38.5 days of real-time gameplay.

$$Game\ Time(day) = \frac{Num\ \ Frames}{10800 * 2 * 24}$$

E. Importance & How to interpret

Now that a few metrics are introduced, the question is, which one is the most suitable? Historically speaking, the HNS was introduced to include the average human performance when comparing the performance of two algorithms and served as an intuitive metric. However, the drawback of only portraying the generality instead of the top performance cannot be ignored. HWRNS has higher requirements for the algorithm to fulfill since it uses the human world records instead of human average performance. It requires the algorithms to pursue a better performance across all the games rather than focus on only a few games. Especially Agent57 made the RL community think over the metrics since it was proposed to be a superhuman agent since it managed to overcome all of the average human performances and therefore reached a mediocre performance. However, its learning efficiency is horrible in terms of taking approximately 100B training frames to gain a SOTA performance [5] which is equal to 19290 days (ca. 52.8 years) real-time game play. Especially 52 years is absurd because it is less than 52.8 years since the birth of the Atari games [1]. If there wasn't even the chance for a human to take this time to reach their records, or in other words if humans reached given performances in less time, how could a DRL agent taking way longer be called superhuman? The results of the performance can be seen in figure 1. The question of whether the Agent57 was now superhuman or not made clear that a new metric is needed in order to evaluate the question properly since the HNS is only representing the average performance neglecting the learning efficiency [1]. It also becomes obvious that the median HNS score is higher than the mean HNS score for the Agent57. This is the case because the mean HNS is not robust against outliers. Agent57 has made a mediocre performance on all games, however has not reached high scores speaking generally which could potentially raise the mean HNS. The fact that Agent57 is taking so long for the SOTA performance is not reasonable when talking about a superhuman performance. Therefore, the most suitable metric when evaluating if an algorithm's performance is superhuman, is currently the learning efficiency and also the HWRNS and HWRB. These metrics require top performances as well as top learning time oriented at the human world records. Nevertheless, it is important to note that still the HNS is used frequently when evaluating algorithm's performances, because a combination of all metrics, HNS, HWRNS, HWRB and learning efficiency, make it possible to see a bigger picture. The metrics have different purposes and it is useful to gain as much information as possible. This is why one should view all of the metrics together in order to get a complex view of the agent's performance. One metric alone may be adapted to a specific benchmark, like the HWRB is only oriented on how many records are reached and not showing to what extent they reached it like the HWRNS. Also only the learning efficiency is not conveying information about the performances. What can be concluded from that is that only one metric is not meaningful for the

superhuman performance evaluation, but also that for future work it might be reasonable to think over a metric which may combine a few of the named information and may evaluate an algorithm in more aspects. It is important to notice how crucial evaluation metrics are in the current context. If there only was the HNS metric and it would be used to evaluate if an algorithm performed superhuman or not, Agent57 would be considered as superhuman. However, as was discussed already above due to its low learning efficiency and with regard to only mediocre performance this is not actually true [1]. Nevertheless, the information of the HNS metric is interesting, but not expressive on its own. When it comes to the latest metrics like HWRNS, HWRB and learning efficiency, there is the new SOTA algorithm called Generalized Data Distribution Iteration that reached higher performance than agent57 as can be seen in figure 1. It will be focussed on the algorithm and its agents GDI-I^3 and GDI-H^3 more in section IV. The inference is that the metric determines what the RL community want to reach, for the HNS metric it was mediocre performance, for the HWRNS it is currently the top performance with regard to human world records. The latest metric HWRB will urge the community to find an agent that can reach all 57 human world records. The current SOTA algorithm GDI has reached 22 human world records. A question arises. What will happen if there will be an agent that can achieve this in a reasonable learning efficiency, e.g GDI as a benchmark within 38,5 days? Would DRL be superhuman on Atari then? This question remains open and may inspire future work.

What was pointed out is that there is the need to discuss or define more clearly what exactly superhuman performance means. Also it was emphasized what impact the metrics have on the algorithms that are coded and on the evaluation of what superhuman agents are. Generally speaking, it is reasonable to use all the metrics, HNS, HWRNS, HWRB and learning efficiency to gain a complex and multilayered opinion. Also developing a metric which may represent a combination of information from the other metrics, or that is generally more expressive on its own may be an interesting idea for future work.

The current benchmark for a superhuman performance is therefore reaching LEARNING EFFICIENCY and TOP PERFORMANCE simultaneously. Even though the superhuman Atari benchmark would be reached in the case of a DRL agent achieving all 57 human world records in reasonable time, there is still the need to think outside the box. What does this superhuman benchmark in Atari games mean for a DRL agent? Has he reached generality then? To what extent can such an algorithm be transferred to real world problems? Is the Atari environment obsolete and there is the need for a benchmark on performances on real world problems, or more complex games? In other words, what is the next step if the superhuman benchmark on Atari is reached?

IV. Generalized data distribution iteration

As it was already elaborated in section III, obtaining a high sample efficiency and superior performance simultaneously has been one of the major challenges of DRL. Especially model-free agents performed pretty well at the cost of a low sample efficiency. One example that was shown so far was the Agent57. It emerges one question: Why is it a challenge for current SOTA algorithms to obtain high sample efficiency and high performance at the same time? In the following this question will be answered in detail. Afterwards, when it is clear what the issue is, it will be discussed what has to be solved in order to overcome this challenge. Finally, the agent GDI will be introduced and is proving as an example on how this challenge can possibly be solved.

1) Data distribution optimization problem: The challenge of gaining a high sample efficiency and superior performance simultaneously in the current context is current for model-free algorithms. A lot of model-free DRL agents which have proven to be SOTA are concerned [1], [5]. The fact that these SOTA algorithms gained high performances was leading to the fact that the underlying strategy was continued. Namely, this strategy can be explained as the idea to traverse all possible conditions and therefore reach remarkable final performance [3]. Under the assumption that there are limitless interactions possible this may work well when it comes to performance. However, in terms of sample efficiency this strategy is critical. Selected behavior policies which are used to generate training data can be represented by the training data distribution [3]. In RL training data distribution is usually controlled by the behavior policy. This means that the final data richness is dependent on the capacity and diversity of the behavior policy [3]. The capacity simply means the amount of different behavior policies which are given, whereas the diversity denotes the amount of behavior policies that are sampled in order to generate new data. The idea of the model-free SOTA algorithms so far was basically that the capacity is increased and the diversity is maximized through randomly sampling of the behavior policies [3]. As the current research has shown [1], this strategy can significantly improve the data richness and guarantee traversal of almost all unseen conditions, which induces better final performance and generalization. The traversal of unseen conditions is representing the exploration of the given conditions. What can be said is that these algorithms are failing the trade-off between Exploration and Exploitation. The problem is the strategy of collecting massive data without prioritizing the value of the data. By strongly focusing on only Exploration, a lot of trials are wasted in order to collect useless, or low-value data which is leading to a low sample efficiency [1]. It is also known as the *data distribution optimization problem*.

2) Data Distribution Optimization: As was indicated already in section IV-1, a lot of SOTA model-free DRL algorithms suffer from the Data distribution optimization problem. The idea is to overcome this problem by also doing Exploitation and weight the value of the given conditions and therefore advance the data distribution. In other words, the solution to the complex phenomenon of the low sample efficiency connected to the exploration and exploitation trade off would be to optimize the data distribution [1]. In a nutshell,

the training data distribution is controlled explicitly whereas the probability of nontrivial conditions being traversed is maximized. Now, one may ask, why is it even necessary to reach a high sample efficiency? It is important to note that the strategy of hard exploration only works when there are (nearly) unlimited interactions guaranteed. In fact, there exist environments, where each interaction is rare and the selection of the behavior policy becomes important [1]. It would be unreasonable to find a way to guarantee a traversal of all unseen conditions. Rather, in order to obtain a Data Distribution Optimization, the probability of the traversal of unseen conditions should be increased via increasing the capacity and diversity of the behavior policy as well as the probability of high-value conditions being traversed via optimizing the selective distribution of the behavior policy should be maximized [1]. In short, exploration and exploitation should be considered equally.

A. Generalized Data Distribution Iteration (GDI)

What can be inferred from section IV-2 is that the sample efficiency of model-free methods can be significantly improved via *data distribution optimization*. As can be observed when the GDI-H^3 agent is introduced, solving the challenge of the sample efficiency problem is not degrading the final performance and therefore the agent is proposing a solution for the well known challenge of simultaneously reaching high sample efficiency and superior performance [3]. To achieve this, a data distribution optimization operator ε is employed in order to iteratively optimize the selective distribution of the behavior policy and thereby the training data distribution is optimized [3].

In more detail, a parametrized policy space is created and indexed by λ which will be defined as the soft entropy space. The behavior policies are sampled from this policy space via a sampling distribution. After that, a meta-learning method is introduced in order to improve the sampling distribution of the behavior policies iteratively. As a consequence, a more detailed exploration and exploitation trade off is reached [3]. Next to that, the training data that is collected by the optimized behavior policies is used for the RL optimization through the operator T. For now the main algorithm handling the data distribution optimization problem will be explained.

B. Notation

The Notation is based on the paper [3]. Λ is an index set, $\Lambda \subseteq R^{\mathrm{k}}.\lambda \in \Lambda$ is an index in Λ. $(\Lambda, B|_\Lambda, P_\Lambda)$ is a probability space, whereas $B|_\Lambda$ is a Borel $\sigma - algebra$ restricted to Λ . In the context of meta Reinforcement learning, Λ could also be described as the set of all possible meta information.

Θ denotes a set of all possible values of parameters. $\theta \in \Theta$ is some specific value of parameters. For each index λ , there is a mapping in between each parameter of θ and λ. It is described as θ_λ. It is showing parameters in θ matching to λ. Concerning linear regression $y = wx, \Theta = \{w \in R^{\mathrm{n}}\}\ and\ \Theta = w$..If λ denotes using only the first half features to make regression, assume $w = (w_1, w_2),\ then\ \theta_\lambda = w_1$. In RL, θ_λ is regarded as a parameterized policy indexed by λ. It is labeled as π_{θ_λ}.

$D = \{d^{\pi}_{\mathrm{p_0}} | \pi \in \delta(A)^{\mathrm{S}},\ p_0 \in \Delta(S)\}$ depicts the set of all state visitation distributions. With regard to the parameterized policies, it is true that $D_{\Lambda,\,\Theta,\,p_0} = \{d^{\pi_{\theta_\lambda}}_{\mathrm{p_0}}\ | \theta \in \Theta, \lambda \in \Lambda\}$, whereas $(\Lambda, B|_\Lambda, P_\Lambda\ )$ is a probability space on Λ inducing a probability space on $D_{\Theta,\,\Lambda,\,p_0}$ given by $P_D(D_{\Lambda_0,\Theta,p_0}) = P_\Lambda(\Lambda_0), \forall \Lambda_0 \in B|_\Lambda$.

x represents one sample, comprising all necessary information for learning. The content of x is dependent on the given algorithm. X is used to represent the set of samples. t denotes a certain training stage t, whereas the parameter $\theta = \theta(t)$, the distribution of the index set $P_\Lambda = P^{(t)}_\Lambda$ and the distribution of the initial state $_0$ are given. The set of samples can therefore be defined as can be observed in the following:

$$\begin{aligned} x^{(t)}_{p_0} &= \bigcup_{d^{\pi}_{p_0} \sim P^{(t)}_D} \{x | x \sim d^{\pi}_{p_0}\} \\ &= \bigcup_{\lambda \sim P^{(t)}_\lambda} \{x | x \sim d^{\pi}_{p_0}, \theta = \theta^{(t)}_\lambda\} \\ &= \bigcup_{\lambda \sim P^{(t)}_\lambda} x^{(t)}_{p_0,\lambda} \end{aligned}$$

C. Model & control of the capacity and diversity of behavior policy

Behavior policies denoted as μ are sampled from a policy space $\{\pi_{\theta_\lambda} | \lambda \in \Lambda\}$ that is parameter-ized by the policy network as well as indexed by Λ.Λ can be seen as the index set. μ's capacity corresponds to the amount of different behavior policies being in the policy space [3]. The policy space is controlled by the base policy's capacity as well as the size of the index set $|\Lambda|$. There are two different sets of parameters, in particular $\Lambda\ and\ \theta$.. By contrast, the diversity equals the amount of behavior policies that were picked then form the given policy space used for generating training data [3]. The control is therefore based on the sampling distribution P_Λ. How is this now helping us concerning the initial problem? When the capacity is successfully determined it is possible to actively control the given data richness in changing the size of the index set and sampling distribution P_Λ. Now, since the control of the data richness was presented, the Data Distribution optimization Operator will be introduced.

D. Data Distribution optimization Operator

T denoting $\theta^{(t+1)} = T(\theta^{(t)}, \{x^{(t)}_{p_0,\lambda}\}_{\ \lambda \sim P^{(t)}_\lambda}\ )$, is a typical optimization operator of RL algorithms [3]. It applies the collected samples to update the parameters for maximizing some function L_{T}. For example, L_{T} may contain the policy gradient and the state value evaluation for the policy-based methods, may contain generalized policy iteration for the value-based methods, may also contain some auxiliary tasks or intrinsic rewards for special designed methods [3]. ε is defined as $P^{(t+1)}_\Lambda = \varepsilon(P^{(t+1)}_\Lambda, \{x^{(t)}_{p_0,\lambda}\}_{\ \lambda \sim P^{(t)}_\lambda}$ and is a data distribution optimization operator [3]. It is using samples $\{x^{(t)}_{p_0,\lambda}\}_{\ \lambda \sim P^{(t)}_\lambda}$ for the purpose of maximizing some function L_ε and to update P_Λ. This process can be described

as $P_\Lambda^{(t+1)} = \arg\max_{P_\Lambda} L_{\varepsilon(\{x_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim P_\lambda^{(t)}})}$ [1]. When P_Λ is parameterized, the notation P_Λ is also used to represent the parameter of P_Λ.

E. Generalized data distribution Iteration

Putting together the control of the capacity and diversity of the behavior policy, as well as the adaptive control of the sampling distribution of the behavior policy using a data distribution optimization with the process of *Generalized policy iteration(GPI)*, a general framework emerges, called GDI. While GPI simply describes the general idea of letting policy-evaluation and policy improvement processes interact, independent of the granularity and other details of the two processes [3]. Almost all reinforcement learning methods are well described as GPI. That is, all have identifiable policies and value functions, with the policy always being improved with respect to the value function and the value function always being driven toward the value function for the policy. If both the evaluation process and the improvement process stabilize, that is, no longer produce changes, then the value function and policy must be optimal [3]. The integrating GDI algorithm can be seen in figure 1.

Algorithm 1 Generalized Data Distribution Iteration

Initialize Λ, Θ, $\mathcal{P}_\Lambda^{(0)}$, $\theta^{(0)}$.
for $t = 0, 1, 2, \ldots$ **do**
  Sample $\{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}}$. {Data Sampling}
  $\theta^{(t+1)} = \mathcal{T}(\theta^{(t)}, \{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}})$. {Generalized Policy Iteration}
  $\mathcal{P}_\Lambda^{(t+1)} = \mathcal{E}(\mathcal{P}_\Lambda^{(t)}, \{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}})$. {Data Distribution Iteration}
end for

Fig. 1. The Generalized Data Distribution Iteration algorithm

F. How to design Data distribution optimization operator

The paper [3] introduces two implementations of GDI, namely GDI-I^3 and GDI-H^3. Let $\Lambda = \{\lambda | \lambda = (\tau_1, \tau_2, \epsilon)\}$. The behavior policy is connected to a soft entropy policy space including policies ranging from very exploratory to purely exploitative. This is why the optimization of the sampling distribution of behavior policy P_Λ can be described as the trade-off between exploration and exploitation [3]. The behavior policy π_{θ_λ} can be defined as $\pi_{\theta_\lambda} = \epsilon * Softmax(\frac{A_{\theta 1}}{\tau_1}) + (1-\epsilon) * Softmax(\frac{A_{\theta 2}}{\tau_2})$. The behavior policy constructs a parameterized policy space, and the index set Λ is constructed by $Lambda = \lambda = (\tau_1, \tau_2, \epsilon)$. For GDI-I^3, $A_{\theta 1}$ *and* $A_{\theta 2}$ are identical advantage functions. Namely, they are estimated by an isomorphic family of trainable variables θ. For GDI-H^3, $A_{\theta 1}$ *and* $A_{\theta 2}$ are different, and they are estimated by two different families of trainable variables which means that θ_1 *and* θ_2 are not identical [3].

V. Evaluation of the performances

Since we've already introduced all of the metrics, it can now be evaluated the performance of GDI-H^3 and GDI-I^3 compared to the other algorithms. GDI-H^3 is the current SOTA algorithm since it has reached the highest scores in the current leading HWRNS and HWRB metric. Additionally, it also reached this in a reasonable Playtime of 38,5 days. The remaining challenge for the DRL community is now to reach all human word records with one agent in reasonable playtime. GDI-I^3 and GDI-H^3, both obtained superhuman performance with remarkable learning efficiency. In the paper [3] it was also proven and shown that generally speaking not using a meta-controller (e.g., the index of behavior policy takes a fixed value) will dramatically degrade performance. This also correlates with other DRL algorithms and might be interesting to note for future work.

	GDI H³	GDI I³	Agent57	Go-Explore	SIMPLE	Rainbow	MuZero
Playtime (Day)	**38,5**	**38,5**	19290	1929	0,19	**38,5**	3858
HWRB	**22**	17	18	15	0	4	19
MEAN HNS (%)	**9620.33**	7810.1	4762.17	4989.31	25.78	873.54	4994.97
MEDIAN HNS (%)	1146.39	832.5	**1933.49**	1451.55	5.55	230.99	**2041.12**
MEAN HWRNS (%)	**154.27**	117.98	125.92	116.89	4,8	28.39	152.10
MEDIAN HWRNS (%)	**50.63**	35.78	43.62	50.50	0,13	4.92	49.80

Fig. 2. Performance of introduced SOTA DRL agents GDI-I^3 & GDI-H^3, Agent57, Go-Explore, SIMPLE, RAINBOW DQN and MuZero in the 57 Atari games of the ALE. The performance is represented by introducing evaluation metrics Playtime (Days), HWRB, Mean HNS(%), Median HNS(%), mean HWRNS(%) and median HWRNS(%). The given values are taken from [[3]]

Is GDI-H^3 superhuman on Atari (according to the given evaluation metrics)? What can be said while evaluating the scores is that, other algorithms such as RAINBOW, Go-Explore and Agent57 have more aimed at reaching a mediocre performance which was also explained by the fact that the HNS metrics was the leading one to evaluate average human performance. It is important to note that Agent57 even outperformed GDI-H^3 in the Median HNS. However as it was already discussed, Agent57 reached this at the cost of more than 52 years of training time. GDI-H^3 therefore aimed more at top performance. It is worth noting that SIMPLE only took about 5 hours. Nevertheless, its performance is very poor compared to the current SOTA algorithms, it has not reached any of the records even. Also worth noting is that Go-Explore has also reached a higher mean HNS score than GDI-H^3, namely 1929 days, which is around 5 years. Go-Explore also outperformed GDI-H^3 in the median HNS, but its value is still lower than the one of Agent57. Still, Agent57 has reached more world records than Go-Explore. Looking at the MuZero algorithm it becomes obvious that this SOTA algorithm outperformed Agent57 in all of the given metrics. MuZero has also reached the highest median HNS and therefore outperformed GDI-H^3 in the median HNS metric. The other scores from MuZero show that the algorithm performed pretty similarly to GDI-H^3 with the difference that GDI-H^3 has reached a high learning efficiency of 38.5 days

of playtime whereas MuZero score of the Playtime is around 10 years. From the historical perspective of RAINBOW it is interesting to see what current SOTA algorithms overcame and have reached. Compared to RAINBOW, GDI-H^3 has reached more than 5 times more human world records in the same game time. Whereas Agent57, Go-Explore and MuZero struggled to reach top performance and learning efficiency equally, GDI-H^3 found a solution to overcome this. In a sense GDI-H^3 is not yet superhuman on Atari since there are still not yet all human world records reached. GDI-H^3 can be seen as an important stepping stone on the way to DRL agents that prove to have generality since it managed compared to other SOTA algorithms to gain high sample efficiency and top performance simultaneously.

VI. Is Deep Reinforcement Learning Really Superhuman on Atari?

As it was shown during the course of this paper, in the past groundbreaking progress in Atari benchmark has been achieved. However, a question arises since Agent57, MuZero and GDI-H^3 already overcame the Atari human benchmark. Is DRL in general really Superhuman on Atari? Is the step to general competency already reached? Yes and No. When just talking theoretically about the defined human benchmark which was outperformed one might say yes, but in reality there are still many challenges in the Atari benchmark that are demonstrating the drawbacks of current RL algorithms [1]. First, these challenges will be elaborated in detail and afterwards there will be given some solutions and suggestions on how to resolve them.

One of the current challenges is that still at least there are 35 human world records that are not broken yet by current SOTA RL algorithms. As a matter of fact DRL algorithms have achieved a human benchmark performance on all the games, but it has not broken all the records and therefore has not reached a Superhuman performance so to say. There is still the challenge of breaking all the human records by DRL algorithms.

Also, as elaborated before a lot of model-free algorithms failed to balance the trade-off between exploration and exploitation leading to lower learning efficiency [1]. What can be inferred from that knowledge is that if it was possible to overcome these challenges a better performance can be expected which could help with the goal of outperforming the human records. One approach would be to adapt the exploration and exploitation balance to a better extent. It is a classic difficult problem in RL algorithms [1]. The design for hard exploration problems may fail the trade-off balance which is leading to a low sample efficiency. Agent57 tries to tackle the problem by training a family of policies from extremely explorative to highly exploitative [5]. Accordingly, GDI suggested employing a data distribution iterator [3] to formulate this procedure and revealed its superiority to the origin process without the data distribution iterator. It may be a promising way to solve this problem.

Another challenge concerns the Planning and Modeling learning. For model-free RL it is extremely difficult when only sparse rewards are given. Especially those algorithms that are without intrinsic rewards are struggling to learn from weak gradient signals [1]. For model-based methods this is not such a problem since they are adopting a world model to plan or replay enhancing the gradient signals. What remains problematic in this scenario is that being dependent on planning is highly unrealistic in terms of modeling reality and will lose generality in specific Atari Games. How could this challenge possibly be resolved? Planning algorithms like Muzero [14] fail when the outcome signals of the planning algorithms become misleading or indistinguishable [1]. The misleading signals may be caused by the model approximation errors which could be solved by employing more advanced deep reinforcement learning methods. The indistinguishable signals are based on the relatively sparse rewards environment like Montezuma Revenge is. This could possibly be solved by a long time of planning. In other words, there is a leading framework needed that guides agents towards a better decision. GDI [3] showed a promising way to combine techniques from Go-Explore and Muzero into the data distribution iteration operator and guide the policy inside an episode, which may solve the low learning efficiency and hard exploration problems.

Another challenge is the learning efficiency. The Agent57 for example would in theory require more than 52 years of game-play to achieve SOTA performance. The learning efficiency is too low to be regarded as superhuman. The high learning efficiency is crucial for aiming at general competency [1]. Additionally, as was argued in [1] 200M training frames which are equal to 38 days would be appropriate for achieving a superhuman agent. But how can this be achieved? Generally when pursuing the DRL community, there have been several independent improvements over years where it still remains unclear if a combination of approaches would be useful as for example the Rainbow agent proved to be. It is more research needed as well as a unifying framework which could make all of the improvements compatible. A unified framework may help to obtain the superhuman agents in the end.

Finally, a challenge is to answer the question of transferability. How good or bad can such a DRL agent with SOTA performance on the Atari benchmark be transferred to other games or environments? In the paper [14] where the MuZero agent is presented, it is also discussed that MuZero has the ability to learn from a model of its environment and use it to successfully plan. As a conclusion, one can say that MuZero's performance is proving a significant advance in RL and the pursuit of general purpose algorithms. It is also shown in the paper [14] that MuZero performed well in chess, Go, Shogi and the Atari games. Therefore, different applications are demonstrated.Its predecessor, AlphaZero, has already been applied to a range of complex problems in chemistry, quantum physics and beyond [14]. The ideas behind MuZero's powerful learning and planning algorithms may pave the way towards tackling new challenges in robotics, industrial systems and other messy real-world environments where the "rules of

the game" are not known [14]. There is a need to make it more transparent for other algorithms how they perform in different games or environments. It could be seen as a benchmark of general performance, for example when there is a more complex environment where the agents are tested, or a real world environment. As was already discussed in the introduction, modern game consoles involve visuals, controls, and a general complexity that rivals the real world. It would be interesting to compare different algorithms on different environments to give a more evaluated view on the general competency and also to define suitable metrics to describe its performance better.

VII. Conclusion

All in all, you can say that even though the defined human benchmark in Atari games was reached by DRL algorithms, they are still far from superhuman performance. It was revealed that there are still open challenges including planning and modeling, hard exploration, human world records, defining suitable metrics for the performance, testing the transferability of the algorithms and low learning efficiency. If in future DRL can achieve this superhuman performance is unclear, however with regard to the breakthroughs of the last few years there is a chance of reaching this stepping stone step by step. One thing is clear, it will be a complex process overcoming all these challenges and developing new approaches. It is definitely more research needed especially in terms of combining approaches as the Rainbow agent for example could show promising results. Furthermore, revisiting the steps that have been done until now, it is useful to keep in mind that the meaning of all these breakthroughs in the Atari games is concerning a general competency. Developing all these agents can be seen as stepping stones towards general competent agents that can be used in several domains. There are already discussions on for which applications suitable DRL algorithms could be used, like for language models, drug design or training robotics to navigate safely [14]. The roads are pioneered and the companies as well as the research is looking forward to new improvements of DRL agents that are proving to have general competency. As was also discussed in section III for future work it may be interesting to find a metric being more expressive on its own, or at least several metrics should be considered to develop an opinion on the agent's performance. With regard to general intelligence of the DRL agents it was also discussed that after the superhuman threshold is reached in Atari games benchmark, it is still questionable to what extent the given algorithm can be transferred to real world problems. It also questions whether the Atari environment may be obsolete and if another environment in addition as a benchmark is needed. Also the train of thought emphasizes that for future work it may be useful to find a way to test if such superhuman Atari benchmark agents have general intelligence in terms of transferring them easily to other problems as discussed in section VI. If there will be once a satisfactory outcome and what time it would take theoretically in order to reach this point remains unclear since it was already elaborated that in the process of improving an approach new challenges and new questions arise.

VIII. Pictures title page

Atari games

University Osnabrück Logo

References

[1] Fan, J. (2022). A Review for Deep Reinforcement Learning in Atari: Benchmarks, Challenges, and Solution. https://doi.org/10.48550/arXiv.2112.04145

[2] Wirbel, E., Moutarde, F. (2019).Is Deep Reinforcement Learning Really Superhuman on Atari? https://doi.org/10.48550/arXiv.1908.04683

[3] Fan, J., Xiao, C. (2022). Generalized Data Distribution Iteration. https://doi.org/10.48550/arXiv.2206.03192

[4] Bellemare, M., Naddaf, Y., Veness, J., Bowling, M. (2013). The Arcade Learning Environment: An Evaluation Platform for General Agents. https://doi.org/10.1613/jair.3912

[5] Badia, A., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, D., Blundell, C. (2020). Agent57: Outperforming the Atari Human Benchmark. https://doi.org/10.48550/arXiv.2003.13350

[6] Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., Silver, D. (2017). Rainbow: Combining Improvements in Deep Reinforcement Learning. https://doi.org/10.48550/arXiv.1710.02298

[7] Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K., Clune, J. (2021). Go-Explore: A New Approach for Hard-Exploration Problems. https://doi.org/10.48550/arXiv.1901.10995

[8] Kaiser, L., Babaeizadeh, M., Miłos, P., Osinski, B., Campbell, R., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., Michalewski, H. (2020). MODEL BASED REINFORCEMENT LEARNING FOR ATARI. https://doi.org/10.48550/arXiv.1903.00374

[9] van Hasselt, H., Hessel, M., Aslanides, J. (2019). When to use parametric models in reinforcement learning? https://doi.org/10.48550/arXiv.1906.05243

[10] Obando-Ceron, J., Castro, P. (2020). Revisiting Rainbow: Promoting more Insightful and Inclusive Deep Reinforcement Learning Research. https://doi.org/10.48550/arXiv.2011.14826

[11] Yang, Z., Bai, S., Zhang, L., Torr, P.(2019). Learn to Interpret Atari Agents. https://doi.org/10.48550/arXiv.1812.11276

[12] Asadi, K., Fakoor, R., Gottesmann, O., Kim, T., Littman, M., Smola, A. (2022).Proximal Iteration for Deep Reinforcement Learning. https://doi.org/10.48550/arXiv.2112.05848

[13] Schmidt, D., Schmied, T. (2021). Fast and Data-Efficient Training of Rainbow: an Experimental Study on Atari. https://doi.org/10.48550/arXiv.2111.10247

[14] Schrittwieser, J., Antonoglou, I., Hubert, Thomas, Simonyan, Karen, Sifre, L., Schmitt, S., Guez,A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., Silver, D. (2020). Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model https://doi.org/10.48550/arXiv.1911.08265