Systematic choice of video game benchmarks in Deep Reinforcement Learning

Elvio Gomes
Institute of Mathematics and Statistics
Federal University of Bahia, UFBA
Salvador, Brazil
elviobg@ufba.br

Marlo Souza
Institute of Mathematics and Statistics
Federal University of Bahia, UFBA
Salvador, Brazil
msouza1@ufba.br

***Abstract*—Deep Reinforcement Learning has gained much attention due to results obtained by its methods to problems of high dimensionality, which were previously intractable or difficult to solve. In this context, video games have been widely used as experimental environments and benchmarks for the evaluation of reinforcement learning algorithms, as well as guiding the development of new methods. Although a lot has been done in Deep Reinforcement Learning since the proposal of its seminal work, little has been discussed about proper methodologies for constructing such evaluation benchmarks. This paper proposes to systematize the choice of video games to be used as a benchmark guaranteeing representativeness and diversity of learning environments based on the use of video game typologies proposed in the area of Game Design Research.**

***Index Terms*—Deep Reinforcement Learning, Benchmark, video games**

I. Introduction

Reinforcement Learning is a set of Machine Learning techniques in which intelligent agents must learn through interactions with the environment. Due to the great complexity of training such methods, the application of traditional methods for Reinforcement Learning, however, was usually constrained to problems of lower dimensionality [1], i.e. problems with states described employing a small number of variables. In this context, the emergence of Deep Reinforcement Learning (DRL), i.e. employing Deep Neural Networks for the problem of Reinforcement Learning, has afforded good performance on complex high-dimensional problems, which were previously intractable by traditional RL methods [1]. The seminal work in this area proposed the Deep Q-Network (DQN)[2], an application of deep neural networks for Q-learning, which was then used to learn how to play Atari 2600 games from raw pixels achieving human-level performance in 49 games.

Many works in the literature have employed digital games as benchmarks for evaluating and comparing different Reinforcement Learning techniques since digital games are dynamic environments, often with difficult to find or multiple simultaneous rewards. These characteristics make digital games a source of challenging and suitable environments for assessing the performance of reinforcement learning algorithms.

A common practice in the literature is the employ a large number of games for benchmarking DRL algorithms, making the evaluation process time-consuming and costly - in the face of the high computational cost of training and executing current DRL algorithms. To our knowledge, there has been no systematic analysis of the benchmarks proposed in the literature nor proposals of methodologies for creating these benchmarks. One such analysis allows us to understand the strengths and biases of digital games benchmarks for evaluating reinforcement learning and the results of empirical comparisons between different methods.

In this work, we propose a methodology for the construction of DRL benchmarks based on digital games. This methodology is based on well-founded principles, considering a typology of games ranging over different dimensions connected to dynamic properties of the learning environment. We validate our methodology by creating a benchmark of Atari games, considering those games commonly used in the literature to evaluate DRL techniques and evaluate the descriptive power of the game typology employed in this work to assess the empirical results of DRL methods.

As a result of our empirical validation, we conclude that an adequate classification of digital games can group similar games and, consequently, environments, reducing the need of using an extensive array of games in empirical evaluations of learning methods. With this, we can optimize the costs of evaluation initiatives and thus potentially facilitate research and the development of new algorithms for the area. Another advantage of this approach is the creation of a balanced representative testbed, removing possible bias in using games with similar environments. A good set of digital games selected by a systematic procedure must be composed of diverse and representative environments, representing better and optimized scenarios for researchers looking to study or develop algorithms that work well in many kinds of environments with the same configuration.

This work is structured as follows: in Section II, we discuss the theoretical background of our work and the main ideas behind Deep Reinforcement Learning; in Section III, we discuss a typology of games based on different characteristics, which will be employed in our work; in Section IV, we present the methodology proposed in this work; in Section V, we describe the empirical validation of our methodology in examining a benchmark of Atari 2600 video games commonly employed in the DRL literature to compare different algorithms. Finally, we present our discussions and final considerations.

II. Background

A common way to formalize reinforcement learning problems is through a Markov Decision Process (MDP) [3]. In this model, an agent at a certain time t executes an action a_t, which influences the environment, taking the agent from the state where he was s_t to a new state s_{t+1} and receiving am associated reward, r_{t+1}. The reward function serves, thus, as an indirect form of supervision and acts as the main learning mechanism of the model.

MDPs are formally described as a 4-tuple (S, A, P, R_t), with a non-empty set of states S containing an initial state s_o, a non-empty set of actions A, a transition function $P(s'|s, a) : S \times A \times S \rightarrow [0, 1]$, which establishes the probability of achieving a state s' after performing action a at the state s, and a reward function $R(s'|s, a) : S \times A \times S \rightarrow \mathbb{R}$.

A solution to an MDP is a policy $\pi : S \rightarrow A$ mapping states to actions that maximizes the expected value of the reward $E\left[\sum_{t=0}^{\infty} \alpha^t R(s_{t+1}|s_t, a_t)\right]$, for some fixed discount factor satisfying $0 \leq \alpha \leq 1$. The policy represents how the agent can behave at a given time and is composed of the probability of selecting an action in a given state. Once an action is selected, a reward signal is obtained as a response from the environment, which is the first form of policy adjustment.

In this context, Reinforcement Learning is the problem of searching such a policy π when the transition function P and reward function R are unknown. An influential technique for RL, in which the reward function $R(\pi)$ is iteratively approximated, is called Q-Learning Algorithm [3]. This technique employs a table with the states and possible actions for the system, called the Q-table, to represent the policy, in which each pair (s, a) of state-action has an associated value called the Q-value that approximates the expected reward of taking action a in the state s.

It is argued that classic RL algorithms do not deal well with environments with high dimensions [2], since it is difficult to represent and extract characteristics from such environments to generate a satisfactory Q-table and compute Q-value. To solve this problem, Mnih et al. [2] in their seminal work proposed Deep Q-Networks (DQN), which employs deep neural networks to approximate the Q-value of an action in a given state. We will discuss DQN in the following.

A. Deep Q-networks

Deep Q-Networks combine Q-Learning with deep learning neural networks, using a neural network instead of the state/action table to define the action to be taken. Although this is the main difference between Q-Learning and DQN, two new mechanisms were inserted called experience replay and the frozen target network [2, 1].

The results obtained by DQN led this technique to become a milestone in the area, instigating the interest in Deep Reinforcement Learning methods. Following DQN, many researchers proposed extensions of the original method, giving rise to a plethora of algorithms such as Double Deep Q-Networks (DDQN)[4], Prioritized replay DDQN (Prior DQN)[5], Dueling DDQN (Duel DDQN)[6], Distributional DQN[7] and Noisy DQN (Noisy Nets)[8], etc.

III. Game Typologies

Unlike other forms of cultural expression such as cinema, music, literature, painting, and architecture, few studies have attempted to characterize games systematically, perhaps due to the difficulty of finding similar characteristics in such a diverse form of expression [9]. In fact, the most common ways of describing games are by comparison to other games, referring to one or more genres, or even comparing to other artistic expressions [10].

Some authors have proposed game typologies using features that represent elements such as the game environment and its dynamics or the way of iterating with the environment so that it is possible to classify games analytically [9]. In this paper, we employ a generic game typology proposed by C. Elverdam and E. Aarseth [10], which can be used to classify both physical and virtual games and is focused on game design characteristics. This typology is based on a set of meta-categories to represent groups of game features with common characteristics, which are further subdivided into dimensions to represent features.

- *Virtual Space*: this meta-category is concerned with the agent's presence in the virtual space where the game takes place if any, and the way the agent interacts and modifies this environment. This meta-category is related to exploration of game environment and how agent will move and interact with that environment. Within this category there are three dimensions of analysis:
  - *Perspective*: describes how the player observes the virtual space of the game. If the player has a complete view of the scenario, the game is said to have an *omnipresent* perspective, and otherwise, if the player has a partial view that depends on the movement towards different scenarios, it is said to have a *vagrant* perspective.
  - *Positioning*: describes how the player can be oriented of their position on the virtual space. If the rules of the game only allow movement to predefined areas, we say the game has an *absolute* position. Otherwise, if the position can be free and even difficult to explain the character's exact position, we say the game has a *relative* position.
  - *Environment dynamics*: describes how the player interacts with the virtual space. If the player can make changes or additions to the environment, we say the environment has *free* dynamics; if the player can make changes to the environment only in predetermined locations, we say the environment has *fixed* dynamics, and if the player cannot make any changes, we say the environment has no dynamics (*none*).
- *Physical Space*: this meta-category is concerned with the player's presence in the physical space where the game

takes place, and for that reason, it is present only in games that have some interaction with this kind of space. Within this category, there are two dimensions of analysis:

  - *Perspective*: describes how the player observes the space of the game. If the player can see the entire physical area of the game, it is said to be *omnipresent*; if movement is necessary to reach new areas of the scenario, it is said to be *vagrant*.
  - *Positioning*: describes the relationship between the player's position in the game and in the physical world. If the player's position in the game is the same as in the physical world, we say the game has *location based* positioning; if the player's position is defined in correlation to other agents in the game, the game is said to have *proximity based* positioning. Finally, the game can have *both* forms of positioning, when it combines the other two factors.

- *External Time*: this meta-category is concerned with the relation of time passage in-game and the time passage in the physical world, and it is related to the length of the game. Within this category, there are two dimensions of analysis:
  - *Teleology*: describes the length of the game. It is said to be *finite* if a game has a set duration and *infinite*, otherwise.
  - *Representation*: describes the relationship between in-game time passage and time in the physical world. If both are similar, the game is said to have *mimetic* time representation, but if it is not related to the passage of time in the physical world, it is said to have *arbitrary* time representation.
- *Internal Time*: this meta-category deals with the effects of the time passage in-game and whether the player's actions can interfere with that. It is related to how the agent must adapt over time and the speed to do it. Within this category, there are three dimensions of analysis:
  - *Haste*: describes the relationship between the passage of time and changes in the game state. Haste is said to be *present* in a game if time passage can change the game state and *absent*, otherwise.
  - *Synchronicity*: describes whether the agents in the game can act synchronously. Synchronicity is *present* in the game if different agents can perform actions simultaneously, and *absent* if actions can be done only one agent per time.
  - *Internal Control*: describes the amount of control the agent has on the passage of time in the game. If players have the power to decide when the next cycle of the game will begin, we say internal control is *present* in the game, and *absent* otherwise.
- *Player Composition*: This meta-category is concerned with how the players come together to form teams. It is related to how many AI agents can exist and how they will be grouped. Within this category, there is only one dimension:
  - *Composition*: describes the organization of the players in the game. If the game does not allow team composition, it is said to be *single-player (1P)*, *two players(2P)*, or *multiplayer*, depending on how many players can participate in the game. If the game allows team structures, i.e. sets of players who play in collaboration, the game is said to be either *single team*, *two teams*, or *multiteam*, depending on the number of teams allowed.
- *Player Relation*: this meta-category is concerned with how players establish relationships among themselves in the game and their influence on the game's goals. Within this category, there are two dimensions of analysis:
  - *Bond* describes how the relationship between players change over time. If these relationships can change during the game, the bond is said to be *dynamic*. Otherwise, it is said to be *static*.
  - *Evaluation*: describes the connection between the players' relationships and the in-game scores or rewards when the game is measured quantitatively. When the game scoring is not influenced by the players' relationships, we say the game has *individual* evaluation. If the scoring is performed on the basis of the players relationships, e.g. if it is defined by the entire team, it is said to have a *team* evaluation. If the scoring combines individual and team evaluations, we say the game has *both* evaluations.
- *Struggle*: this meta-category is concerned with the relationship between players and game goals, as well as the way challenges are delivered to the player during the game execution. Within this category, there are two dimensions of analysis:
  - *Goals*: describes the victory conditions imposed by the game. If they are unique and immutable, the game is said to have *absolute* goals. Otherwise, they are *relative*.
  - *Challenge*: describes how the players encounter challenges and oppositions in their gameplay. If the challenges are predefined and always repeated in the same way, they are said to be *identical*. If the challenges are predefined but presented with a degree of randomness, they are said to be based on *instance*. Otherwise, if all challenges are delivered by game agents autonomously, we say the game has *agent*-based challenges.
- *Game State*: this meta-category is concerned with mechanisms that change the way the player relates to the game. It is related to whether the agent can interact with the environment in different ways according to the game state. Within this category, there are two dimensions of analysis:
  - *Mutability*: describes how the agent can affect the state of the game. The game has *temporal* mutability if agents or the player can affect the state of the game for a limited time. We say the game has *finite* if these

changes exist and last longer, until the end of the game, it is said to be *finite*, and if these changes are permanent and will exist not only in that game but in all matches or moment after that it is said to be *infinite*. Otherwise, it kind of change can not exists, and it is said to be *none*.

- *Savability*: describes whether the agent has the ability to save and restore the game state. It is said to be *unlimited* if it can be saved and restored at any moment, *conditional* if it can only be saved and restored in certain circumstances or moments, or if the game has no option of saving its state, it is said to have no savability (*none*).

In Table I, we summarize Elverdam et al.'s [10] multidimensional game typology, discussed above.

TABLE I
ELVERDAM ET AL.'S [10] GAME TYPOLOGY

Meta-category	**Dimension**	**Value**
Virtual Space	Perspective	omnipresent
		vagrant
	Positioning	absolute
		relative
	Environment Dynamics	free
		fixed
		none
Physical Space	Perspective	omnipresent
		vagrant
	Positioning	location-based
		proximity-based
		both
External Time	Teleology	finite
		infinite
	Representation	mimetic
		arbitrary
Internal Time	Haste	present
		absent
	Synchronicity	present
		absent
	Internal Control	present
		absent
Player Composition	Composition	one player
		two players
		multiplayer
Player Relation	Bond	static
		dynamic
	Evaluation	individual
		team
		both
Struggle	Goals	absolute
		relative
	Challenge	identical
		instance
		agent
Game State	Mutability	temporal
		finite
		infinite
		none
	Savability	unlimited
		conditional
		none

IV. A METHODOLOGY FOR VIDEO-GAME BENCHMARKS FOR DRL

In this work, we study the use of a game typology to systematically create a benchmark for Deep Reinforcement Learning Algorithms based on digital games. With this typology, we aim to propose a methodology for designing a comprehensive benchmark that can be used to develop and compare the performance of Deep Reinforcement Learning algorithms, presented in Fig. 1.

According to an appropriate game typology, our proposed methodology consists of selecting appropriate digital games based on a diversity of game features to compose a benchmark. It consists of 5 steps, presented below:

- *Step 1*: the creation of a benchmark must begin with an initial selection of candidates of a given platform. In this first step, selecting a representative set of games for the platform is crucial to ensure diversity of learning environment and game characteristics to compose the final benchmark.
- *Step 2*: once the games to be used are chosen, it is necessary to choose the typology that will classify them.
- *Step 3*: after choosing the typology to be used, all games must be analyzed and classified based on the selected typology according to its characteristics.
- *Step 4*: once all the games have been classified, games with similar characteristics must be grouped.
- *Step 5*: in the last step, games with repetitive characteristics, i.e. that were in the same group, must be removed randomly, creating a more concise testbed with balanced environments and representation.

V. EXPERIMENTS

In this section, we evaluate the proposed methodology by performing a statistical analysis of the descriptive power of the adopted game typology and benchmark selection methodology for predicting the performance of Deep Reinforcement Learning algorithms. In other words, we evaluate whether a set of different DRL methods achieve statistically similar performance on games grouped in the same categories, according to our methodology and the adopted typology. This study aims to validate whether a benchmark created using our proposed methodology can be used interchangeably with a broader, non-optimized benchmark.

We will conduct a statistical analysis of the performance of six Deep Reinforcement Learning Algorithms, namely DQN, DDQN, Prior DQN, Dueling DDQN, Distributional DQN, and Noisy DQN, on a benchmark of 55 Atari 2600 games commonly used in the literature. All of the games employed in our study have been previously used in the literature to measure the performance of DRL algorithms [1, 11]. We choose this benchmark since the extensive selection of games provides a good diversity of environments for our statistical analysis and allows us to explore how much computational resources can be saved using a principled methodology to construct a leaner benchmark.

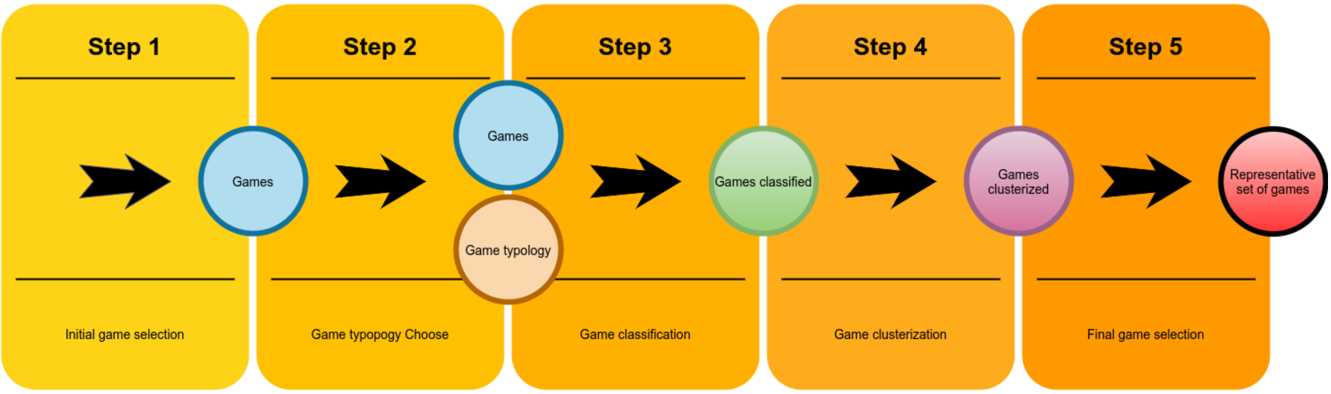

Fig. 1. Methodology steps

A. Game Classification

The classification of the initial selection of games, according to the adopted game typology, was performed by direct observation of gameplay by a human judge. The results of the classification are presented in Table VII. We illustrate the analysis and classification process by discussing the characteristics of the game Space Invaders.

The whole game takes place on only one screen, with aliens enemies up screen and player mobile defence cannon at the bottom. The objective is to destroy all enemies before they go down or shoot the player's cannon. Because the player can see all virtual game space in only one screen, we can categorize Virtual Space Perspective as omnipresent. As the player's cannon can move to left and right, this moves changes player positioning. However, these places cannot be used to explain exactly where the player is in an easy way. Although the absolute position of the cannon can be defined by the axis of movement and pixels on the screen, small variations coming from inputs may not be able to change the position of the cannon significantly. For example, in a situation where the cannon is behind the barricades, small changes will still leave the player behind that barricade, and its position will be relative to that barricade. A different situation from games like Chess, where we can say that White Queen starts in square D1, and all changes in that piece will change its position totally. So we have a relative positioning. Another action that can be done by a player is to shoot to destroy enemies and barricades. These barricades are pre-defined and are the only place of environment that can be changed by the player. Because of this, we can say that the environment dynamics is fixed.

In the game, there is no time representation that reminds time passage in the real world. Enemies, cannons and projectiles can move in a time that we can see, which creates its own time representation. In that case, External Time representation is considered arbitrary.

From time to time, aliens shoot and go down. If aliens reach the bottom, the player loses, so Internal Time Haste is classified as present because it changes the game state, defeating the player. Player and aliens can shoot and move at the same time, so we can say that they can perform actions simultaneously. Thus, synchronicity is present in the game's Internal Time. The player and the enemies do not change the relationship with the game environment, such as receiving power-ups or changing the way they interact with the environment, at any time that changes the game state, so we can say that Game State Mutability is Absent.

Regarding the challenges to achieve the victory, the aliens always move in the same movement choreography, but the enemy shoot is not predictable. It presents some form of randomness, so the Struggle Challenge is based on Instance. Since that game is played only by one player at a time, the Player Composition is one player.

Although the adopted typology has many meta-categories and dimensions, we have noticed that the selected Atari 2600 games have similar characteristics, in the sense that all games analyzed present *finite external time teleology*, for example. The set of meta-categories and dimensions without variation that was not considered in this work was i) Physical Space; ii) External Time: Teleology; iii) Internal Time: Haste, Synchronicity and Internal Control; iv) Struggle: Goals; Game State: Salvability; and v) Player Relation: Bond. We believe this is the case due to the technological limitations of the Atari 2600 console, which resulted in only a small group of dimensions with variation among the games within the typology.

B. Analytical tools and Results

After classification of all 55 games, we obtained 31 groups of games with the same characteristics. Our aim now is to validate the hypothesis that by creating a benchmark by randomly choosing only one game out of each group, the resulting benchmark will have the same separability power as the original benchmark. In other words, we wish to validate the hypothesis that the performance of Deep Reinforcement Learning algorithms is statistically equivalent within all games in the same group.

For that, we use hypothesis testing [12] to check whether different games have similar performance distribution among

DRL algorithms and analyze the difference between games from the same group and different groups.

In our validation, we employ the absolute performance values of six DRL algorithms based on DQN [11]. These algorithms are: Deep Q-Networks, Double Deep Q-Networks (DDQN), Prior DQN, Duel DQN, Distributional DQN, and Noisy DQN. To obtain these values, the authors used the average scores of the agent evaluated during training for 200M frames. For every 1M steps in the environment, they suspended learning and evaluated the agent for 500K frames. All results published by authors are available in Table VI.

With these results, we conducted two analyses: at the first one, we used only DQN results and hypothesis Student's t-test [12] on each pair of games to analyze the similarity between games from the same group and different groups. This experiment uses 49 games in 30 groups.

To accommodate the fact that different games have different scoring systems, score ranges, and variable difficulty, we normalize the score based on the performance of a random agent, i.e. an agent that chooses the following action at random. To obtain the values for the random agent, we used the mean value of 100 games played by the agent, where a random action was performed for every sixth frame. This number of frames is equivalent to 10 Hz, which is about the fastest time that a human can press a 'fire' button [1]. All values obtained by the random agent are present in Table VI with other agent results.

Notice that our normalized score does not involve any human actor, which could introduce confounding factors based, for example, on the semiotic decisions of the game, and induce bias in the evaluation metric. The normalized score employed in this work can be computed in the following way represented in equation 1. Where *norm* is the score normalized, *Score* is the absolute score value[2][11], *Random* is the score obtained by random agent, and ϵ is a discount constant. In this work, we employed $\epsilon = 0.00000001$, a small value to ensure that no division by zero will affect the results.

$$norm = (|Score| - |Random|)/(|Score| + |Random| + \epsilon) \tag{1}$$

After the normalization phase, we have normalized scores per game in an acceptable range that can be used to make the hypothesis Student's t-test to verify the similarity between a pair of normalized game scores. After hypothesis tests, we take the mean of the obtained p-values and their variance from the same game group and different groups. To games from same group we have p-value mean = 0.016 and variance = 0.005 and for games from different groups mean = 0.032 and variance = 0.017. These results are in Table II and results group by group are in Table III.

The second analysis was made by grouping results from different algorithms trained to play the same game and using this sample to make hypothesis tests to whether the achieved performance of each pair of games was similar and then analyze the similarity between games from the same group and different groups. This experiment uses 52 games in 31 groups.

TABLE II
MEANS AND VARIANCES OF P-VALUES FROM HYPOTHESIS TESTS IN EXPERIMENT 1

	p-value mean	**p-value variance**
Same Group	0.30	0.09
Different Groups	0.36	0.14

TABLE III
HYPOTHESIS TESTS FROM GAME BY GROUPS IN EXPERIMENT 1

# Games	**Games**	**p-value mean**	**p-value variance**
2	Hero Krull	0.125	0
4	Battle zone Chopper command James Bond Up and Down	0.385	0.08
2	River raid Robot tank	0.0008	0
2	Centipede Space invaders	0.14	0
2	Atlantis Ms Pacman	1.76E-05	0
6	Assault Beam rider Demon attack Enduro Star gunner Time pilot	0.31	0.07
4	Asteroids Gravitar Road runner Zaxxon	0.38	0.145
6	Asterix Boxing Gopher Kung fu master Name this game Seaquest	0.28	0.13

In this experiment, we used the same normalization described previously in equation 1. After the normalization phase, we have 6 normalized scores per game in an acceptable range that can be used to make hypothesis tests game by game. These values are one to each of the DRL algorithms, DQN, DDQN, Prior DQN, Duel DQN, Distributional DQN, and Noisy DQN. The hypothesis tests used were Student's t-test for parametric distributions and Wilcoxon [13] for non-parametric, to decide which should be used, we make a normality check using the Shapiro-Wilk test [13].

After hypothesis testing we take p-values mean and variance from same game group and different groups. To games from same group we have p-value mean = 0.056 and variance = 0.016 and for games from different groups mean = 0.070 and variance = 0.024. These results are also in Table IV results group by group are also in Table V.

VI. DISCUSSION

In the first experiment, we observed a more significant p-value mean from games of the same group than from different

TABLE IV
MEANS AND VARIANCES OF P-VALUES FROM HYPOTHESIS TESTS IN EXPERIMENT 2

	p-value mean	p-value variance
Same Group	0.045	0.027
Different Groups	0.095	0.033

TABLE V
HYPOTHESIS TESTS FROM GAME BY GROUPS IN EXPERIMENT 2

# Games	Games	p-value mean	p-value variance
2	Hero Krull	0.028	0
3	Battle zone Berzerk Chopper command	0.017	0.0001
2	Centipede Space Invaders	1.94e-05	0
2	Atlantis Ms Pacman	0.028	0
9	Assault Beam rider Demon attack Enduro Phoenix Solaris Star gunner Time pilot yYars revenge	0.042	0.024
4	Asteroids Gravitar Road runner Zaxxon	0.020	0.002
6	Asterix Boxing Gopher Kung fu master Name this game Seaquest	0.079	0.055

groups like exposed in Table II. These differences suggest that games from the same group have statistically more similar performances than games from different groups. Also, we obtained a smaller p-value variance within games of the same group, indicating greater stability of the results.

While experiment 1 indicates that DQN achieves similar performance for games within the same group, by using only one method, our results can be biased to specific characteristics of DQN. As such, in experiment 2, we employed six different DRL methods to validate our observations.

The second experiment has similar results compared with the first one but with a more expressive statistic difference. Tests with the same game groups have smaller p-value means and variances compared with games from different groups like exposed in Table IV. Again, these results suggest DRL algorithms achieve more similar performances in games of the same groups than in different groups. In fact, by analyzing Table IV, we can see that DRL algorithms achieve statistically significant similar performances for games in almost all the analyzed groups.

Results in groups, such as group #7 in Table V, which obtained a high p-value (0.079), can indicate that some games can have some features that are uncommon to other games of that group and that these features are being untreated by this game typology.

VII. CONCLUSION

This paper proposed a methodology for analyzing and choosing video game-based benchmarks for DRL by selecting the most diverse possible set of games of different groups and removing those with similar features. Also, with this methodology, it is possible to analyze whether some testbed lacks a diversity of game features and, thus, may be biased towards specific learning environment characteristics.

Using the same games as in the original DQN work[1] we show that a commonly used benchmark in the area can be reduced by removing games with repetitive features. To prove that games with similar features have similar performance with respect to DRL algorithms, we compared the results of DQN and its variations in a large benchmark of games and compared these results in light of our typology, showing that games within the same group have statistically similar performance and are more similar than games from different groups. As for the testbed size, it is possible to see that in the case analyzed, we were able to reduce the number of games from 55 to 31, a reduction of 45%, which can improve new research time without loss of diversity.

As for the representativeness of our benchmark, according to our typology, it is possible to note that the set of Atari 2600 games used does not contemplate all possible features and groups, and a lot of the games have similar characteristics, with some groups made of 6 and 9 games. These results indicate that new benchmarks with greater diversity are necessary and may further help to guide empirical comparisons between methods and improving our understanding of DRL methods in various environments. In that sense, our work provides a systematic process to create such benchmarks.

It is demonstrated that a game typology can use game features to group similar games, improving research time and decreasing associated computational costs. It implies the necessity to choose a good benchmark to guide new developments in the area.

In future work, we intend to extend the current game typology adding new meta-categories and dimensions with agent-centric attributes such as the dimensionality of the state and action space, determinism, average game length and reward distribution and removing less-useful meta-categories to the study of DRL, such Physical Space, thus creating a typology more related to intelligent agents instead of game design. We also intend on extending our analyses for other methods of Reinforced Learning, no only those methods based on DQN.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level

control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[3] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[4] H. Hasselt, "Double q-learning," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010, pp. 2613–2621.

[5] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2016.

[6] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2016.

[7] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," 2017.

[8] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," 2019.

[9] E. Aarseth, S. M. Smedstad, and L. Sunnanå, "A multidimensional typology of games," in *DiGRA Conference*, 2003.

[10] C. Elverdam and E. Aarseth, "Game classification and game design: Construction through critical analysis," *Games and Culture*, vol. 2, no. 1, pp. 3–22, 2007. [Online]. Available: https://doi.org/10.1177/1555412006286892

[11] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," *CoRR*, vol. abs/1710.02298, 2017. [Online]. Available: http://arxiv.org/abs/1710.02298

[12] P. G. Hoel, *Introduction to Mathematical Statistics*, 4th ed., ser. Probability & Mathematical Statistics. John Wiley & Sons Inc, 1971.

[13] W. Conover, *Practical nonparametric statistics*, 3rd ed., ser. Wiley series in probability and statistics 1999: 9. Wiley, 1999.

VIII. Appendix

TABLE VI
RESULTS FROM DIFFERENT AGENTS. *RESULTS FROM MNIH ET AL. [1]. **RESULTS FROM HESSEL ET AL. [11] ***RESULTS BY AUTHOR

Game	DQN*	DQN**	DDQN**	Prior DDQN**	Duel DDQN**	Distrib DQN**	Noisy DQN**	Random***
Alien	3069	1620	3747.7	6,648.60	4,461.40	12,944.00	18,347.50	326.4
Amidar	739.5	978	1793.3	2,051.80	2,354.50	1,267.90	1,608.00	24
Assault	3359	4280	5393.2	7,965.70	4,621.00	5,909.00	5,198.60	144.9
Asterix	6012	4359	17356.5	41,268.00	28,188.00	400,529.50	12,403.80	162
Asteroids	1629	1364.5	734.7	1,699.30	2,837.70	2,354.70	4,814.10	153.2
Atlantis	85641	279987	106056	427,658.00	382,572.00	273,895.00	329,010.00	2088
Bank heist	429.7	455	1030.6	1,126.80	1,611.90	1,056.70	1,323.00	11.4
Battle zone	26300	29900	31700	38,130.00	37,150.00	41,145.00	32,050.00	2450
Beam rider	6846	8627.5	13772.8	22,430.70	12,164.00	13,213.40	12,534.00	172.48
Berzerk		585.6	1225.4	1,614.20	1,472.60	1,421.80	837.3	131.5
Bowling	42.4	50.4	68.1	62.6	65.5	74.1	77.3	21.06
Boxing	71.8	88	91.6	98.8	99.4	98.1	83.3	-48.15
Breakout	401.2	385.5	418.5	381.5	345.3	612.5	459.1	2.3
Centipede	8309	4657.7	5409.4	5,175.40	7,561.40	9,015.50	4,355.80	1483.71
Chopper command	6687	6126	5809	5,135.00	11,215.00	13,136.00	9,519.00	354
Crazy climber	114103	110763	117282	183,137.00	143,570.00	178,355.00	118,768.00	1050
Demon attack	9711	12149.4	58044.2	70,171.80	60,813.30	110,626.50	24,950.10	202.2
Double dunk	-18.1	-6.6	-5.5	4.8	0.1	-3.8	-1.8	-24
Enduro	301.8	729	1211.8	2,155.00	2,258.20	2,259.30	1,129.20	0.4
Fishing derby	-0.8	-4.9	15.5	30.2	46.4	9.1	7.7	-94.71
Freeway	30.3	30.8	33.3	32.9	0	33.6	32	7.21
Frostbite	328.3	797.4	1683.3	3,421.60	4,672.80	3,938.20	583.6	48.6
Gopher	8520	8777.4	14840.8	49,097.40	15,718.40	28,841.00	15,107.90	24
Gravitar	306.7	473	412	330.5	588	681	443.5	23
Hero	19950	20437.8	20130.2	27,153.90	20,818.20	33,860.90	5,053.10	1233.25
Ice hockey	-1.6	-1.9	-2.7	0.3	0.5	1.3	-2.1	-16.35
James Bond	576.7							3.5
Kangaroo	6740	7259	12992	14,492.00	14,854.00	12,909.00	12,117.00	97
Krull	3805	8422.3	7920.5	10,263.10	11,451.90	9,885.90	9,061.90	82.48
Kung fu master	23270	26059	29710	43,470.00	34,294.00	43,009.00	34,099.00	9
Montezuma revenge	0	0	0	0	0	367	0	0
Ms pacman	2311	3085.6	2711.4	4,751.20	6,283.50	3,769.20	2,501.60	356.4
Name this game	7257	8207.8	10616	13,439.40	11,971.10	12,983.60	8,332.40	570
Phoenix		8485.2	12252.5	32,808.30	23,092.20	34,775.00	16,974.30	13.6
Pitfall		-286.1	-29.9	0	0	-2.1	-18.2	-90.71
Pong	18.9	19.5	20.9	20.7	21	20.8	21	-21
Private eye	1788	146.7	129.7	200	103	15,172.90	3,966.00	1249.39
Qbert	10596	13117.3	15088.5	18,802.80	19,220.30	16,956.00	15,276.30	193.25
River raid	8316							211
Road runner	18257	39544	44127	62,785.00	69,524.00	63,366.00	41,681.00	79
Robot tank	51.6	63.9	65.1	58.6	65.3	54.2	53.5	3.95
Seaquest	5286	5860.6	16452.7	44,417.40	50,254.20	4,754.40	2,495.40	82.4
Skiing		-13062.3	-9021.8	-9,900.50	-8,857.40	-14,959.80	-16,307.30	-30000
Solaris		3482.8	3067.8	1,710.80	2,250.80	5,643.10	3,204.50	74.6
Space invaders	1976	1692.3	2525.5	7,696.90	6,427.30	6,869.10	2,145.50	154
Star gunner	57997	54282	60142	56,641.00	89,238.00	69,306.50	34,504.50	380
Tennis	-2.5	12.2	-22.8	0	5.1	23.6	0	-23.94
TIme pilot	5947	4870	8339	11,448.00	11,666.00	7,875.00	6,157.00	493
TUtankham	186.7	68.1	218.4	87.2	211.4	249.4	231.6	0
Up and Down	8456							529.8
Venture	3800	163	98	863	497	1,107.00	0	0
Video pinball	42684	196760.4	309941.9	406,420.40	98,209.50	478,646.70	270,444.60	753.97
Wizard of wor	3393	2704	7492	10,373.00	7,855.00	15,994.50	5,432.00	40
Yars revenge		18089.9	11712.6	16,451.70	49,622.10	16,608.60	9,570.10	2021.68
Zaxxon	4977	5363	10163	13,490.00	4,055.80	2,394.90	9,491.70	0

TABLE VII
CLASSIFIED AND GROUPED GAMES

Game	Virtual Space			Internal Time		Ext Time	Game State	Struggle	Comp.
	Perspective	**Pos**	**Env**	**Haste**	**Sync**	**Rep**	**Mutability**	**Challenge**	**Players**
Hero Krull	Vagrant	Relative	Fixed	None	Present	Arbitrary	None	Instance	1P
Tutankham	Vagrant	Relative	Fixed	Present	Present	Arbitrary	Temporal	Instance	1P
Skiing	Vagrant	Relative	None	None	None	Mimetic	None	Instance	1P
Private eye	Vagrant	Relative	None	None	Present	Arbitrary	Temporal	Identical	1P
Battle zone Berzerk Chopper command James Bond Up and Down	Vagrant	Relative	None	None	Present	Arbitrary	None	Instance	1P
Montezuma	Vagrant	Relative	None	None	Present	Mimetic	None	Identical	1P
Alien	Vagrant	Relative	None	Present	Present	Arbitrary	Temporal	Identical	1P
River raid Robot Tank	Vagrant	Relative	None	Present	Present	Arbitrary	None	Instance	1P
Venture	Vagrant	Relative	None	Present	Present	Arbitrary	None	Identical	1P
Pitfall	Vagrant	Relative	None	Present	Present	Mimetic	None	Identical	1P
Crazy climber	Vagrant	Absolute	Fixed	None	Present	Arbitrary	None	Instance	1P
Breakout	Omni	Relative	Fixed	None	Absent	Arbitrary	None	Identical	1P
Wizard of wor	Omni	Relative	Fixed	None	Present	Arbitrary	Temporal	Instance	2P
Kangaroo	Omni	Relative	Fixed	None	Present	Arbitrary	None	Instance	1P
Frostbite	Omni	Relative	Fixed	None	Present	Arbitrary	None	Identical	1P
Video pinball	Omni	Relative	Fixed	None	Present	Mimetic	None	Identical	1P
Centipede Space Invaders	Omni	Relative	Fixed	Present	Present	Arbitrary	None	Instance	1P
Bowling	Omni	Relative	None	None	None	Mimetic	None	Identical	1P
Atlantis Ms pacman	Omni	Relative	None	None	Present	Arbitrary	Temporal	Instance	1P
Fishing derby	Omni	Relative	None	None	Present	Arbitrary	None	Instance	2P
Assault Beam rider Demon attack Enduro Phoenix Solaris Star gunner Time pilot Yars revenge	Omni	Relative	None	None	Present	Arbitrary	None	Instance	1P
Freeway	Omni	Relative	None	None	Present	Arbitrary	None	Identical	2P
Asteroids Gravitar Road runner Zaxxon	Omni	Relative	None	None	Present	Arbitrary	None	Identical	1P
Double dunk	Omni	Relative	None	None	Present	Mimetic	Temporal	Instance	1P
Amidar	Omni	Relative	None	None	Present	Mimetic	Temporal	Identical	1P
Ice hockey	Omni	Relative	None	None	Present	Mimetic	None	Instance	2P
Pong	Omni	Relative	None	None	Present	Mimetic	None	Identical	1P
Tennis	Omni	Relative	None	None	Present	Mimetic	None	Instance	1P
Asterix Boxing Gopher Kung fu master Name this game Seaquest	Omni	Relative	None	Present	Present	Arbitrary	None	Instance	1P
Bank heist	Omni	Relative	None	Present	Present	Arbitrary	None	Identical	1P
Qbert	Omni	Absolute	None	None	Present	Arbitrary	None	Instance	1P