LevDoom: A Benchmark for Generalization on Level Difficulty in Reinforcement Learning

Tristan Tomilin [1], Tianhong Dai [2], Meng Fang [1], Mykola Pechenizkiy [1]
[1] Eindhoven University of Technology, The Netherlands [2] Imperial College London, United Kingdom
t.tomilin@tue.nl, tianhong.dai15@imperial.ac.uk, m.fang@tue.nl, m.pechenizkiy@tue.nl

***Abstract*—Despite the recent success of deep reinforcement learning (RL), the generalization ability of RL agents remains an open problem for real-world applicability. RL agents trained on pixels may completely be derailed from achieving their objectives in unseen situations with different levels of visual changes. However, numerous existing RL suites do not address this as a primary objective or lack consistent level design of increased complexity. In this paper, we introduce the LevDoom benchmark, a suite containing semi-realistic 3D simulation environments with coherent levels of difficulty in the renowned video game Doom, designed to benchmark generalization in vision-based RL. We demonstrate how our benchmark reveals weaknesses of some popular Deep RL algorithms, which fail to prevail in modified environments. We further establish how our difficulty level design presents increasing complexity to these algorithms.**

***Index Terms*—reinforcement learning, generalization, vizdoom**

I. Introduction

Though deep reinforcement learning has made immense leaps forward over the past decade in the domain of video games [1]–[5], robotics [6]–[9], and a lot of other applications [10]–[13], generalization remains one of the most fundamental challenges for RL [14]. Modern algorithms require large amounts of collected experience to function in the applicable domain [15]. However, this may be unavailable or expensive. Generalizing to unseen scenarios thus proves challenging for embodied AI, as the effective strategies learned in the training environment may not later be adequate. It has previously been shown that slight visual modifications on pixel-based observations from Atari games may completely disrupt a well trained policy [16]. Whereas humans are able to seamlessly generalize across similar tasks, this competence is still predominantly absent in RL agents, who tend to instead become exceedingly specialized to the environments which they encounter during training [17]. The lack of generalizability makes self-learning systems unreliable for real-world applications (e.g., robotics, automation, healthcare and finance) where robustness is crucial [18]. Targeting generalization is thus particularly vital as it endorses the AI to thrive in unencountered conditions, which is much desired for artificial general intelligence, rather than just solving individual problems.

Proper generalization benchmarks are vital for RL research, as they provide means of comparing the performance of methods and techniques on unseen environments with little effort. Some such recently proposed platforms are repurposed on 3-dimensional game engines such as ViZDoom [19], DeepMind Lab [20], and MineRL [4], and are thus able to facilitate a realistic perspective from a first-person point of view, which pave the way to more lifelike means of applicability. However, numerous existing RL suites like ALE [1], OpenAI Gym [21], RL-Lab [22], ViZDoom [19], and DeepMind Lab [20] do not implicitly regard generalization as a primary focus or lack a broad and consistent difficulty level design [23]–[26]. Other related benchmarks for generalization research [25]–[29] primarily define difficulty as an implicit property of the game's mechanics (e.g., stronger enemies or more complex goals), which is generally employed to increase the challenge for human players, but may be inadequate for measuring generalization of RL agents.

In this paper, we present the LevDoom Benchmark[1] , a suite containing over 50 semi-realistic simulation environments in four scenarios, adhering to a coherent notion of difficulty across multiple levels, designed to research and evaluate generalization in vision-based RL. LevDoom is based on ViZDoom [30], a semi-realistic 3D world offering virtual embodiment and egocentric perception. Real-world applications are subject to constant environmental changes, presenting a great variety of visual input to RL agents. This raises the demand for a policy which is able to generalize across such instability. To this end, we visually modify each environment within a scenario by changing e.g., surface textures, entity types, shapes and sizes, and modes or rendering. Compared to existing benchmarks oriented towards generalization, LevDoom explicitly distinguishes between individual modifications to enable researchers to discern how RL agents respond to particular unencountered environment alterations. Unlike previous benchmarks [25], [26], [28], [29], [31], we propose to express difficulty in terms of the number of visual modification types within the environment. We posit that this approach creates a coherent concept of difficulty, which enables to better quantify generalizability.

The contributions of our work are three-fold:

1) We introduce the LevDoom benchmark, comprised of four scenarios with environments of increasing difficulty, to meet the growing needs for proper evaluation mechanisms for generalizable RL agents.
2) We employ three well-known algorithms (DQN [32], Rainbow [33], and PPO [6]) to train baseline models

[1]All environments and code are open-source and can be found at https://github.com/TTomilin/LevDoom.

TABLE I: Scenario properties

Scenario	Success Metric	Action Space	Episode Timeout	Enemies	Weapon
Defend the Center	Frames Alive	ATTACK, TURN_LEFT, TURN_RIGHT	1300	✓	✓
Health Gathering	Frames Alive	MOVE_FORWARD, TURN_LEFT, TURN_RIGHT	2100	✗	✗
Seek and Slay	Kill Count	ATTACK, MOVE_FORWARD, TURN_LEFT, TURN_RIGHT	1250	✓	✓
Dodge Projectiles	Frames Alive	MOVE_LEFT, MOVE_RIGHT, SPEED	2100	✓	✗

on lower level environments of each of the scenarios, and compare their performance on more challenging environments of higher difficulty.

3) We establish how our proposed mechanism of level difficulty indeed poses an increasing challenge, and demonstrate to what degree the popular algorithms fail to perform on slightly modified environments of our benchmark.

II. Related Work

A. Benchmarks in RL

Rapid progress has been made in both the 2D and immersive 3D simulation environment domain [1], [2], [14], [21]–[23], [31], [34]–[37]. Multiple research platforms and benchmarks have been based on existing repurposed game engines. **DeepMind Lab** [20] is built upon Quake III Arena [38], facilitating creative tasks, navigational challenges, and intelligence tests on visual cues. **Project Malmo** [39] is based on Minecraft, challenging the agent with navigation, survival, collaboration and problem solving. **ViZDoom** [30] is based on the classical FPS video game Doom, facilitating learning from raw visual information in a semi-realistic world. These platforms do not, however, address generalization as the primary objective or lack difficulty levels.

Several benchmarks in RL address generalization [23]–[25], [40], [40]. The **Sonic Benchmark** [23] measures generalizability by separating a very limited number of levels of the Sonic the Hedgehog™video game for training and evaluation. The **General Video Game Artificial Intelligence** (GVGAI) [40] competition framework poses the problem of training agents that can play a wide and unlimited range of games. The **DeepMind Memory Task Suite** [24] is comprised of a diverse set of memory tasks to evaluate the memory and generalization of agents. **Meta-World** [25] explores how meta-learning algorithms can quickly learn new tasks when meta-trained on a task distribution of continuous control environments. **The Procgen Benchmark** [26] measures sample efficiency and generalization across 16 environments, enabling the algorithmic creation of a near-infinite supply of highly randomized levels and content with procedurally generated content PCG [41]. **MazeExplorer** [42] assesses generalization in navigation and exploration. **CRLMaze** [43] presents a non-stationary object-picking task, subject to constant environmental changes. Compared to previous generalization benchmarks, LevDoom particularly targets visual modifications. This is crucial when learning from pixel data to better distinguish the impact of particular variations.

B. Difficulty Levels

A benchmark with levels of difficulty is anything but new. The **BabyAI** platform [31] comprises a suite of 19 scenarios of increasing difficulty for grounded language learning, measuring difficulty in the combination of competencies required for solving a task. The improved version of ALE [28] supports combinations of game mode and difficulty level pairs called *flavors*. **Obstacle Tower** [29] consists of 100 PCG environments of increasing difficulty in the third-person perspective in, with low-level control and high-level planning problems. We take inspiration from these works in designing difficulty levels by combining visual modification types. This enables to determine which visual alterations are most impactful and provides a broader evaluation mechanism. Previous benchmarks mainly increase implicit difficulty, which only impacts game dynamics which we posit to be less vital for generalization of embodied agents that learn from pixels.

III. LevDoom Benchmark

We introduce the scenarios and environments of the LevDoom benchmark in detail, and explain the corresponding difficulty levels, evaluation protocols and limitations.

A. Environments

With LevDoom we aim to build a benchmark, that facilitates evaluating generalizability in modified semi-realistic 3D simulation environments. The initial version of the benchmark consists of four manually designed scenarios (Defend the Center, Health Gathering, Seek and Slay, and Dodge Projectiles), each with environments of progressing difficulty levels. Every scenario is designed with a particular narrow objective, but nevertheless establishing high skill requirements.

Each environment in the benchmark is a modified version of an original map from ViZDoom [19], a flexible RL research platform for learning from raw visual information, based on the classical FPS video game Doom. We design new environments by modifying the *Internal WAD*[2] of the original scenario. The environments are of pseudorandom nature, which manifests in the randomized behaviour of enemies, fluctuating damage inflicted by attacks, and spawning locations of items, enemies, and the agent. All enemies have 1 health. The Doom environments is run at a frame rate of 35 FPS. The weapon[3] and heads-up display (HUD) are visually rendered, whereas the crosshair, particles, and decals (materials projected onto

[2]A WAD file is a game data file used by FPS games running on the original Doom engine.

[3]The weapon is only rendered in environments in which the agent is granted one.

TABLE II: Example environments per difficulty level of each scenario.

Scenario	Environments				
	Level 0	Level 1	Level 2	Level 3	Level 4
Defend the Center	 Default	 Gore	 Stone Wall + Flying Enemies	 Resized Flying Enemies + Mossy Bricks	 Complete
Health Gathering	 Default	 Resized Kits	 Slime + Obstacles	 Lava + Supreme + Resized Agent	 Complete
Seek and Slay	 Default	 Shadows	 Obstacles + Resized Enemies	 Red + Obstacles + Invulnerable	 Complete
Dodge Projectiles	 Default	 Barons	 Revenants	 Flames + Flaming Skulls + Mancubus	 Complete

existing surfaces) are not. Each episode is terminated after a predetermined number of frames (see Table I).

To provide an evident recognition of environments in literature, we adopt an evidently discernible nomenclature, in which the name of an environment represents the modified attribute(s) it incorporates. The names of environments may thus overlap across scenarios. We consider this naming convention feasible for the initial iteration of the benchmark, which does not require naming environments with more than three combined modifications, therefore not excessively lengthening the names.

1) Observation Space: Doom is by far not a complete information game, since at a single point in time, the agent can spatially occupy only one location of the entire environment and observe a portion of its surroundings. We use a 4:3 in-game resolution, which grants a 90 degree field of view (FoV). The observation space $\mathcal{S}$ is an image of the environment from the first person perspective. This image is rendered in a 160 × 120 resolution with 3-channels of 8-bit values in RGB.

2) Action Space: We use a multi-discrete action space $\mathcal{A}$ which varies across scenarios, but remains fixed among environments within a scenario. Similarly to ViZDoom [19], we do not use the full set of actions from the Doom game for simplicity. The available actions per scenario are presented in Table I.

3) Rewards: We design two base rewards $\mathcal{R}$ for each environment E, closely dependent on the success metric of the scenario S. The reward r_t at every time step t is calculated as follows:

- Defend the Center & Seek and Slay

$$r_t = k_t, \tag{1}$$

where k is the number of enemies eliminated.

- Health Gathering & Dodge Projectiles

$$r_t = F, \tag{2}$$

where F is the base reward for surviving a frame.

TABLE III: Comparison of existing generalization benchmarks with LevDoom.

Benchmark	3D	Input Dim.	Environments	Modifications
Alchemy [44]	✓	96x72x3	(Generated)	Shapes, colors, topologies, game logic
Sonic [23]	✗	320x224x3	11	Levels from multiple Sonic video games
Distracting Control [45]	✗	448x448x3	14	Textures, camera pose, video backgrounds
Obstacle Tower [46]	✓	168x168x3	(Generated)	Lighting, textures, floor plan, room layout
Procgen [26]	✗	64x64x3	16	Level layout, game assets, entity spawns
MazeExplorer [42]	✓	320x240x3	(Generated)	Multiple maps, spawns, textures, keys
CRLMaze [43]	✓	320x240x3	28	Light, textures, object shapes, colors
LevDoom	✓	160x120x3	53	Textures, rendering modes, entity types & sizes, view height

B. Difficulty Levels

We assign a difficulty level $d \in \{0, \ldots, 4\}$ to each environment E based on the number of modifications. The unmodified *default* environment of every scenario is level $d = 0$. The level of an environment is in accordance with the number of difficulty attributes it incorporates (e.g., a level 3 environment is comprised of three modification types), with the exception of the *complete* environments (level $d = 4$), which includes all scenario specific difficulty attributes. We further define the set of all environments of difficulty d in a scenario S as $\mathcal{E}_d \subset \mathcal{E}$. Apart from levels 0 and 4, there is no fixed number of environments per level. We depict one environment of each level per scenario in Table II. We take Defend the Center as an example to illustrate in Table IV how difficulty levels are determined by combining modifications. The modification types of other scenarios may differ.

We modify the environment characteristics with the goal of changing their visual appearance. Most of the variations we introduce (e.g., textures, entity size, agent height, rendering mode) would not increase difficulty for human players. However, some modifications (e.g., entity type, obstacles) may have a subtle effect on the game mechanics. The proposed modifications include the following:

- Introducing new enemy and item types.
- Rendering enemies and items in a different shape, size, or style.
- Applying noisier textures, which increase the challenge of distinguishing the relevant enemies or items from the background.
- Adding decorations to the environment, which either act as obstacles by hindering the navigation of the agent, or confuse the agent as being potential relevant targets.
- Varying the height of the agent, which vertically shifts the plain of view.

Doom additionally incorporates a configurable in-game difficulty setting, which determines the speed and aggressiveness of enemies, the factor of damage taken by the player, and further characteristics, which are not relevant to our environments. We set this parameter to a value of 3 from the range of 1-5 for all environments.

C. Evaluation Protocol

An agent is trained in a multi-task setting on a scenario S on all environments of lower levels $d^{tr} = \{0, 1\}$. We thus define the training environment set as $\mathcal{E}^{tr} = \mathcal{E}_0 \cup \mathcal{E}_1$. The agent is then evaluated on higher difficulty levels $d^{te} = \{2, 3, 4\}$. We hence define the set of test environments as $\mathcal{E}^{te} = \mathcal{E}_2 \cup \mathcal{E}_3 \cup \mathcal{E}_4$.

D. Limitations

Some environment modifications in our benchmark may affect game dynamics in addition to visual disparity. For example, among the modifications outlined in Table IV, *Obstacles* and *Entity Type* have such and impact. Hence, two environments with the same modification types from a scenario might have different game dynamics.

The benchmark only consists of four scenarios, thus merely addressing a few competencies, whereas there are numerous possibilities for designing additional scenarios, which may target navigation, memory, spatial reasoning, or exploration. This leaves the door open for expanding the benchmark.

For simplicity and ease of training, we restricted the action spaces of scenarios, mainly keeping only actions which are crucial for accomplishing the established objective. Allowing the agent to perform more actions may increase overall performance when employing more powerful methods.

IV. Experiments

In this section we evaluate the generalizability of agents on environments with different levels of difficulty using the LevDoom benchmark. We describe the agent models we used for running the generalization experiments. We outline the experimental protocols and present the results of the experiments.

A. Setup

We use five seeds per experiment to control the pseudo-random nature of the environments. We train each model for 10M environment iterations, and evaluate it after every 100K iterations for 30 episodes on each holdout environment. We select a total of seven test environments per scenario: three environments from levels $d = 2$ and $d = 3$ each, and the single final level $d = 4$ environment *complete*. The performance on a test environment is determined by the mean of the score across the five runs, according to the scenario specific metric. During preprocessing we downscale the input to a 84 × 84 pixel grayscale image. Neglecting RGB channels as input to the model is computationally cheaper and we hypothesize that grayscale images is more advantageous for generalization, since a great number of visual modifications in our environments incorporate color.

TABLE IV: Combining environment modifications of the Defend the Center scenario.

Level	Environment	Textures	Obstacles	Entity Size	Entity Rendering	Entity Type	Entity Speed
0	Default						
1	Gore		✓				
	Mossy Bricks	✓					
	Stone Wall	✓					
	Fuzzy Enemies				✓		
	Resized Enemies			✓			
	Fast Enemies						✓
	Flying Enemies					✓	
2	Gore + Mossy Bricks	✓	✓				
	Resized Fuzzy Enemies			✓	✓		
	Stone Wall + Flying Enemies	✓				✓	
3	Resized Flying Enemies + Mossy Bricks	✓		✓		✓	
	Gore + Stone Wall + Fuzzy Enemies	✓	✓		✓		
	Fast Resized Enemies + Gore		✓	✓			✓
4	Complete	✓	✓	✓	✓	✓	✓

1) Agent Models: We use three popular algorithms for reinforcement learning in high-dimensional environments: **DQN** [32], **Rainbow** [33], and **PPO** [6]. We use the algorithm implementations from the RL platform Tianshou [47]. For off-policy methods we use a replay buffer of size 100K instead of 1M to lower the algorithm's memory consumption.

2) Reward Shaping: In addition to the base rewards outlined in Equations 1 and 2, we heuristically extend the reward functions by including additional components to enhance the feedback to the agent. We thus calculate the reward r_t at every time step t per scenario as follows:

- Defend the Center

$$r_t = k_t - m_t - h_t, \tag{3}$$

  where $k = 1.0$ indicates the number of enemies eliminated, $m = 0.1$ marks using ammo, and $h = 0.1$ signals taking damage from enemy attacks.

- Health Gathering

$$r_t = F - p_t + h_t, \tag{4}$$

  where $F = 0.01$ is the base reward for surviving a frame, $p = 1.0$ stands for picking up poison, and $h = 1.0$ indicates acquiring a health item.

- Seek and Slay

$$r_t = s_t + c\left\|l_t - l_{t-k}\right\|_2^2, \tag{5}$$

  where $s = 1.0$ indicates the number of enemies slayed, l marks the coordinates of the agent's location in the environment, and $k = 5$ determines the number of time frames in the past from which the covered distance is calculated. Note that the distance component is not considered before k iterations of an episode have passed.

- Dodge Projectiles

$$r_t = F - h_t, \tag{6}$$

  where $F = 0.01$ is the base reward for surviving a frame and $h = 0.1$ indicates the penalty of taking damage from enemy attacks.

3) Hardware: The GPU used for running our experiments is an MSI GeForce GTX 1080 Ti, with 11GB of RAM, 3584 CUDA cores, and a compute capability of 6.1. The CPU is an Intel i7-7700 CPU with 8 hyperthreads, and a processing speed of 3.60GHz. There is 32GB of RAM.

B. Results on Generalization

In Figure 1, we present the evaluation curves of DQN, Rainbow and PPO on the LevDoom benchmark, and in Table V we display the averaged scores of the final ten evaluations. We can observe that PPO had the best performance in the Dodge Projectiles scenario, whereas Rainbow emerged superior in all the rest. Out of the 28 evaluated environments, our DQN agent outperformed other algorithms in only 2 environments, whilst PPO surmounted in 10, and Rainbow in 16. It can be noticed that PPO had very poor performance on the Seek and Slay scenario. We hypothesize that the reason for this is the lack of a replay buffer to reuse collected experience, which would be beneficial in this scenario.

In Figure 2, we further establish how our difficulty level design presents increasing complexity. To this end we select the Rainbow agent, as it reaped the best performance, and display its aggregated evaluation results per level. We can indeed observe that the performance in all scenarios drops as the difficulty level increases. It can be noticed that the between-level performance gaps are similar across scenarios, which indicates a coherent difficulty level design, on which LevDoom particularly emphasizes. The curve of the level 4 *complete* environment appears rather flat for most scenarios, which suggests that training on lower level environments did not provide the Rainbow agent with sufficient generalization capability to prevail the hardest environment.

V. Conclusion

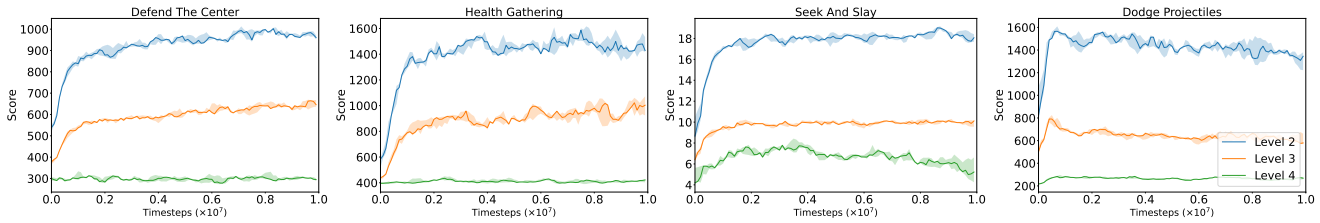
Training proficient agents, who are able to generalize across environments, currently remains one of the greatest challenges in reinforcement learning. To aid the community in grappling with this challenge, in this paper, we introduced and

(a) Defend The Center (b) Health Gathering (c) Seek and Slay (d) Dodge Projectiles

Fig. 1: A comparison between DQN, Rainbow and PPO. We train the agents on all environments of levels 0 and 1, and evaluate them on environments of higher levels. The success metrics (*Score*) of scenarios are outlined in Table I. The solid line is the median value across five seeds. The upper bound and lower bound are the 25$^{\text{th}}$ and 75$^{\text{th}}$ percentile, respectively.

TABLE V: Quantitative comparison between DQN, Rainbow and PPO. We train the agents on all environments of levels 0 and 1, and evaluate them on environments of higher levels. The success metrics (*Score*) of scenarios are outlined in Table I. The results are shown as the mean value ± standard deviation of the last 10 evaluation epochs across five seeds. The highest scores are in **bold**.

Defend the Center		Stone Wall Flying Enemies	Resized Fuzzy Enemies	Gore Mossy Bricks	Gore Stone Wall Fuzzy Enemies	Resized Flying Enemies Mossy Bricks	Fast Resized Enemies Gore	Complete	Average
	DQN	904.55±53.67	1219.17±53.67	506.77±50.41	387.08±14.44	737.25±38.60	481.73±50.30	283.06±17.58	645.66±24.23
	Rainbow	**1097.28±53.94**	**1289.76±47.63**	530.09±30.61	**405.61±15.21**	**885.41±32.03**	**637.24±45.86**	302.73±11.63	**735.45±11.94**
	PPO	599.21±89.71	884.85±111.19	**805.29±122.22**	336.39±26.14	556.14±51.38	409.08±31.14	**454.75±76.36**	577.96±43.63
Health Gathering		Supreme Poison	Slime Obstacles	Shaded Stimpacks	Poison Resized Shaded Kits	Obstacles Slime Stimpacks	Lava Supreme Resized Agent	Complete	Average
	DQN	1064.74±81.69	1858.67±107.20	706.26±96.38	754.04±72.14	845.76±82.23	485.14±21.55	412.53±15.15	875.31±38.54
	Rainbow	**1237.85±76.80**	**1946.02±63.50**	**1164.90±168.80**	**1244.31±150.91**	**1196.30±215.84**	466.48±20.44	412.86±12.46	**1095.49±43.85**
	PPO	1058.99±55.76	1925.52±65.14	917.85±72.91	621.60±34.24	992.44±99.70	**573.22±25.95**	**415.89±11.43**	929.36±22.67
Seek and Slay		Blue Shadows	Obstacles Resized Enemies	Invulnerable Blue	Blue Mixed Resized Enemies	Red Obstacles Invulnerable	Resized Shadows Red	Complete	Average
	DQN	7.71±1.23	20.78±0.98	18.80±0.83	9.72±0.59	16.38±0.77	**1.21±0.31**	4.10±0.64	11.24±0.32
	Rainbow	**12.18±1.18**	**22.48±0.85**	**19.81±0.57**	**11.82±0.69**	**16.96±0.76**	1.17±0.18	**5.77±0.94**	**12.88±0.30**
	PPO	4.81±2.26	4.83±2.35	9.63±0.94	3.93±1.05	8.79±0.92	0.43±0.32	4.61±1.16	5.29±1.17
Dodge Projectiles		City Resized Agent	Revenants	Barons Flaming Skulls	City Arachnotron	Flames Flaming Skulls Mancubus	Resized Agent Revenants	Complete	Average
	DQN	1365.14±125.73	1300.88±118.50	1819.79±102.54	**670.23±46.02**	900.34±102.38	590.97±44.57	266.15±14.23	987.64±46.26
	Rainbow	1318.47±156.04	1040.21±139.49	1753.30±92.05	575.86±63.89	773.08±81.02	503.46±36.82	260.92±12.80	889.33±59.51
	PPO	**1944.84±49.46**	**1802.23±72.72**	**1918.54±53.27**	584.61±25.45	**1018.26±61.51**	**749.13±50.95**	**287.53±7.39**	**1186.45±21.33**



(a) Defend The Center (b) Health Gathering (c) Seek and Slay (d) Dodge Projectiles

Fig. 2: Performance of Rainbow on unseen evaluation environments of increasing difficulty. The agent is trained on all environments of levels 0 and 1, and evaluated on environments of higher levels. The solid line is the median value across five seeds. The upper bound and lower bound are the 25th and 75th percentile, respectively.

openly released LevDoom, a novel benchmark for assessing generalization on visually modified environments with levels of difficulty, and used it to evaluate and compare popular RL algorithms. Experimental results on four different scenarios demonstrate that our level design of combining visual modifications increasingly hampers the performance of three popular RL algorithms on unseen environments. We have demonstrated that unseen environments including all modification types of textures, in-game entities and further attributes completely derail the agents from achieving the limited established objectives. This design of level difficulty makes the benchmark essential for evaluating generalization in RL. We expect this benchmark to facilitate the design of more generalizable algorithms and methods.

REFERENCES

[1] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[2] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser *et al.*, "Starcraft ii: A new challenge for reinforcement learning," *arXiv preprint arXiv:1708.04782*, 2017.

[3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[4] W. H. Guss, B. Houghton, N. Topin, P. Wang, C. Codel, M. Veloso, and R. Salakhutdinov, "Minerl: A large-scale dataset of minecraft demonstrations," *arXiv preprint arXiv:1907.13440*, 2019.

[5] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[7] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[8] M. Fang, C. Zhou, B. Shi, B. Gong, J. Xu, and T. Zhang, "Dher: Hindsight experience replay for dynamic goals," in *International Conference on Learning Representations*, 2018.

[9] M. Fang, T. Zhou, Y. Du, L. Han, and Z. Zhang, "Curriculum-guided hindsight experience replay," *Advances in Neural Information Processing Systems*, 2019.

[10] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[11] M. Fang, Y. Li, and T. Cohn, "Learning how to active learn: A deep reinforcement learning approach," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 595–605. [Online]. Available: https://www.aclweb.org/anthology/D17-1063

[12] Y. Xu, M. Fang, L. Chen, Y. Du, J. T. Zhou, and C. Zhang, "Deep reinforcement learning with stacked hierarchical attention for text-based games," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 16 495–16 507.

[13] Y. Xu, L. Chen, M. Fang, Y. Wang, and C. Zhang, "Deep reinforcement learning with transformers for text adventure games," in *IEEE Conference on Games (CoG)*, 2020, pp. 65–72.

[14] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, "Quantifying generalization in reinforcement learning," in *International Conference on Machine Learning*, 2019, pp. 1282–1289.

[15] B. Eysenbach, S. Asawa, S. Chaudhari, S. Levine, and R. Salakhutdinov, "Off-dynamics reinforcement learning: Training for transfer with domain classifiers," *arXiv preprint arXiv:2006.13916*, 2020.

[16] S. Gamrian and Y. Goldberg, "Transfer learning for related reinforcement learning tasks via image-to-image translation," in *International Conference on Machine Learning*, 2019, pp. 2063–2072.

[17] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, "A study on overfitting in deep reinforcement learning," *arXiv preprint arXiv:1804.06893*, 2018.

[18] G. Lample and D. S. Chaplot, "Playing FPS games with deep reinforcement learning," *arXiv preprint arXiv:1609.05521*, 2016.

[19] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," in *IEEE Conference on Computational Intelligence and Games*, 2016, pp. 1–8.

[20] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik *et al.*, "Deepmind lab," *arXiv preprint arXiv:1612.03801*, 2016.

[21] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[22] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[23] A. Nichol, V. Pfau, C. Hesse, O. Klimov, and J. Schulman, "Gotta learn fast: A new benchmark for generalization in rl," *arXiv preprint arXiv:1804.03720*, 2018.

[24] M. Fortunato, M. Tan, R. Faulkner, S. Hansen, A. P. Badia, G. Buttimore, C. Deck, J. Z. Leibo, and C. Blundell, "Generalization of reinforcement learners with working and episodic memory," *arXiv preprint arXiv:1910.13406*, 2019.

[25] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning*, 2020, pp. 1094–1100.

[26] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman, "Leveraging procedural generation to benchmark reinforcement learning," in *International Conference on Machine Learning*, 2020, pp. 2048–2056.

[27] B. Beyret, J. Hernández-Orallo, L. Cheke, M. Halina, M. Shanahan, and M. Crosby, "The animal-ai environment: Training and testing animal-like artificial cognition," *arXiv preprint arXiv:1909.07483*, 2019.

[28] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. Hausknecht, and M. Bowling, "Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents," *Journal of Artificial Intelligence Research*, vol. 61, pp. 523–562, 2018.

[29] A. Juliani, A. Khalifa, V.-P. Berges, J. Harper, E. Teng, H. Henry, A. Crespi, J. Togelius, and D. Lange, "Obstacle tower: A generalization challenge in vision, control, and planning," *arXiv preprint arXiv:1902.01378*, 2019.

[30] M. Wydmuch, M. Kempka, and W. Jaśkowski, "Vizdoom competitions: Playing Doom from pixels," *IEEE Transactions on Games*, vol. 11, no. 3, pp. 248–259, 2018.

[31] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. H. Nguyen, and Y. Bengio, "Babyai: A platform to study the sample efficiency of grounded language learning," *arXiv preprint arXiv:1810.08272*, 2018.

[32] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[33] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[34] Y. Liang, M. C. Machado, E. Talvitie, and M. Bowling, "State of the art control of atari games using shallow reinforcement learning," *arXiv preprint arXiv:1512.01563*, 2015.

[35] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[36] H. Caselles-Dupré, L. Annabi, O. Hagen, M. Garcia-Ortiz, and D. Filliat, "Flatland: a lightweight first-person 2-d environment for reinforcement learning," *arXiv preprint arXiv:1809.00510*, 2018.

[37] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[38] id Software, "Quake III Arena," 1999, accessed: 2021-05-21.

[39] M. Johnson, K. Hofmann, T. Hutton, and D. Bignell, "The malmo platform for artificial intelligence experimentation." in *International Joint Conference on Artificial Intelligence*, 2016, pp. 4246–4247.

[40] D. Perez-Liebana, J. Liu, A. Khalifa, R. Gaina, J. Togelius, and S. Lucas, "General video game ai: a multi-track framework for evaluating agents," *Games and Content Generation Algorithms*, 2018.

[41] S. Risi and J. Togelius, "Increasing generality in machine learning through procedural content generation," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 428–436, 2020.

[42] L. Harries, S. Lee, J. Rzepecki, K. Hofmann, and S. Devlin, "Mazeexplorer: A customisable 3d benchmark for assessing generalisation in reinforcement learning," in *IEEE Conference on Games*, 2019, pp. 1–4.

[43] V. Lomonaco, K. Desai, E. Culurciello, and D. Maltoni, "Continual reinforcement learning in 3d non-stationary environments," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 248–249.

[44] J. X. Wang, M. King, N. Porcel, Z. Kurth-Nelson, T. Zhu, C. Deck, P. Choy, M. Cassin, M. Reynolds, F. Song *et al.*, "Alchemy: A benchmark and analysis toolkit for meta-reinforcement learning agents," *arXiv preprint arXiv:2102.02926*, 2021.

[45] A. Stone, O. Ramirez, K. Konolige, and R. Jonschkowski, "The distracting control suite–a challenging benchmark for reinforcement learning from pixels," *arXiv preprint arXiv:2101.02722*, 2021.

[46] M. Pleines, J. Jitsev, M. Preuss, and F. Zimmer, "Obstacle tower without human demonstrations: How far a deep feed-forward network goes with reinforcement learning," in *IEEE Conference on Games*, 2020, pp. 447–454.

[47] J. Weng, H. Chen, D. Yan, K. You, A. Duburcq, M. Zhang, H. Su, and J. Zhu, "Tianshou: A highly modularized deep reinforcement learning library," *arXiv preprint arXiv:2107.14171*, 2021.