DOJO: A BENCHMARK FOR LARGE SCALE MULTI-TASK REINFORCEMENT LEARNING

Dominik Schmidt
TU Wien
e11809917@student.tuwien.ac.at

ABSTRACT

We introduce *Dojo*, a reinforcement learning environment intended as a benchmark for evaluating RL agents' capabilities in the areas of multi-task learning, generalization, transfer learning, and curriculum learning. In this work, we motivate our benchmark, compare it to existing methods, and empirically demonstrate its suitability for the purpose of studying cross-task generalization. We establish a multi-task baseline across the whole benchmark as a reference for future research and discuss the achieved results and encountered issues. Finally, we provide experimental protocols and evaluation procedures to ensure that results are comparable across experiments. We also supply tools allowing researchers to easily understand their agents' performance across a wide variety of metrics.

1 INTRODUCTION

Recent research in reinforcement learning has yielded an abundance of learning algorithms able to learn to perform complex sequential decision-making tasks without human supervision (Mnih et al., 2015; Silver et al., 2016; Hessel et al., 2018). While these results are remarkable, in most cases, each trained agent can only perform a single, specific task and new agent instances need to be trained from scratch for every subsequent task (Hessel et al., 2019; Vithayathil Varghese & Mahmoud, 2020). Additionally, prior work has shown that learned policies are often brittle and can break down when environment features are even slightly perturbed (Raileanu et al., 2020). This can also be observed within research on generalization where policies learned through RL have been shown to generalize poorly to previously unseen tasks (Cobbe et al., 2019; 2020). The same issues may also hinder the adoption of RL for practical applications, where robustness and the ability to adapt within a rich and dynamically changing world are essential.

For these challenges, the field of multi-task reinforcement learning (MTRL) has emerged as a potential solution. MTRL is concerned with developing single RL agents that can perform a whole spectrum of tasks without needing to be retrained. Apart from practical advantages, these methods aim to yield policies that are substantially more robust, generalize better to other tasks within the training distribution, and transfer to entirely different tasks (Team et al., 2021). Additionally, multi-task learning agents can exhibit better data efficiency since experience can be shared across tasks (Espeholt et al., 2018).

Standardized benchmarks such as ImageNet (Deng et al., 2009) and the Arcade Learning Environment (ALE) (Bellemare et al., 2013; Machado et al., 2018) have been a cornerstone of research in many areas of machine learning. They allow researchers to reliably compare entirely dissimilar approaches to important problems and serve as a measure of research progress. While many such benchmarks exist for general RL, we believe that most of these are not well suited for research in multi-task RL. If that is indeed the case, the continued reliance on them could greatly impede further research progress.

Thus, in this paper, we present *Dojo*, a new benchmark aiming to fill this gap. Dojo consists of close to 2000 different video game tasks drawn from several existing libraries. In the implementation, special attention was paid to performance, configurability, and ease of use. In particular, Dojo closely adheres to the gym environment API (Brockman et al., 2016) and can, for the most part, be used as a drop-in replacement for existing benchmarks while also supporting more specific use cases. In this work, we particularly focus on applications in multi-task learning, but we believe that the benchmark will also be a useful tool in related areas such as curriculum learning, transfer learning, as well as other open-ended learning settings.

In Section 2 we give an overview of the current issues present in multi-task RL and discuss how recent methods have attempted to address them. We then list available benchmarks and discuss how they are used to (but frequently fail to) evaluate approaches to these issues. In Section 3, we describe the specifics of our own benchmark, compare it to others, and detail how it avoids the listed pitfalls. In Section 4, we present experiments performed to support the design of the benchmark itself, better understand the effect of knowledge transfer within Dojo, and establish a preliminary multi-task baseline.

2 Related Work

2.1 Multi-task reinforcement learning

Multi-task RL is uniquely affected by a number of issues not present in general single-task RL. This section provides a brief overview of these and discusses existing solutions from the literature.

Task interference. When learning multiple tasks jointly, tasks may interfere with each other for a variety of reasons. *Negative knowledge transfer* refers to the case where a behavior learned in one task negatively affects the agent's learning progress in another – for example by impeding exploration of the environment (Vithayathil Varghese & Mahmoud, 2020). *Catastrophic forgetting* can also destabilize learning, particularly in the continual learning case (Fedus et al., 2020). Finally, differences in the return magnitudes of different tasks which in turn cause differences in the error magnitudes can cause an imbalance of learning progress across tasks (Hessel et al., 2019; Schaul et al., 2021). Since the return distribution is generally unknown and non-stationary, naive normalization methods are insufficient. PopArt normalization (van Hasselt et al., 2016), originally introduced for the single-task setting, normalizes targets online but preserves the original unnormalized value function by modifying the last layer of the value network. Return-based scaling (Schaul et al., 2021) is an algorithm-agnostic method that instead rescales the error based on a normalization factor derived from statistics of the environment. PopArt-IMPALA (Hessel et al., 2019) combines the IMPALA algorithm from (Espeholt et al., 2018) with PopArt normalization and applies it to the multi-task setting.

Training speed. Although large computational costs and long training times have also been a trait of much of recent work in general RL (Badia et al., 2020; Kapturowski et al., 2019; Schrittwieser et al., 2020; Hessel et al., 2018), these issues affect multi-task RL to an even greater extent. Evaluation of MTRL methods necessitates experiments with benchmarks consisting of larger numbers of tasks, since MTRL-specific issues may only be observable there. Thus, training speed has been a focus area of some recent work in MTRL. In particular, IMPALA (Espeholt et al., 2018), an actor-critic agent focusing on scalability, training throughput, and computational efficiency, has been used as a research platform and baseline in some recent work (Hessel et al., 2019; Luo et al., 2020).

2.2 Benchmarks in reinforcement learning

In this section, we provide an overview of popular existing benchmarks in RL and discuss their suitability for the purpose of MTRL research. We focus on visual, game-based environments since these are most closely related to our benchmark. For a more comprehensive list of RL environments commonly used to study generalization (many of which are also relevant for multi-task learning) see Kirk et al. (2021).

Arcade Learning Environment (ALE). Since its introduction in 2013, the ALE (Bellemare et al., 2013; Machado et al., 2018) has been one of the primary benchmarks used in RL research. It consists of 57 Atari games, each of which exposes a discrete action space with up to 18 actions and uses RGB images as observations. While the diversity of games in the ALE has been regarded as a benefit for evaluating general RL methods, the minimal task overlap and stark differences between games make it less suitable for MTRL research.

Procgen. This suite of 16 games specifically developed for RL research, replicates some of the games in the ALE, but adds procedural level generation and improved configurability and performance (Cobbe et al., 2020). Like its predecessor CoinRun (Cobbe et al., 2019), Procgen has been used primarily for research in generalization (Hilton et al., 2020; Raileanu et al., 2020; Laskin et al., 2020).

Meta Arcade. While similar to the ALE, Meta Arcade enables researchers to modify many aspects of each provided game, define new games, and allows for a degree of procedural generation within games. The benchmark is intended for research in meta and multi-task learning and includes 24 predefined games (Staley et al., 2021).

gym-retro. This library serves as an interface between classic video games and RL algorithms by exposing various emulators as RL environments conforming to the gym-API (Nichol et al., 2018). The gym-retro package ships with around 1000 game integrations included, but due to per-game differences in action, observation, and reward spaces can be cumbersome for use in multi-task learning.

DeepMind Lab. DeepMind Lab (Beattie et al., 2016) is a library providing access to a collection of 3d game-based tasks that has been used in similar settings as the ALE. It includes 10 predefined environments along with other contributed environments such as the 30 DMLab-30 tasks.

MineRL. MineRL is an open-ended environment based on the game of Minecraft and includes more than 20 predefined tasks. It has been employed in competitions on sample efficient RL with human demonstrations (Shah et al., 2021; Guss et al.).

We find that many of the existing benchmarks suffer from several issues that make them unsuitable for research in MTRL. Ideally, MTRL benchmarks should contain a large number of tasks including both clusters of similar and dissimilar tasks. This would allow researchers to evaluate how well their algorithms can generalize across related tasks, avoid interference between unrelated tasks, and scale to large numbers of tasks. In contrast, most existing benchmarks include only few, largely unrelated tasks with little skill overlap (such as the ALE or the different Procgen games). There, MTRL approaches have little benefit over conventional single-task methods, while suffering from additional complexity. Furthermore, the limited number of tasks makes it difficult to investigate how MTRL methods scale with the number of tasks. Environments that use procedural generation to increase diversity (such as CoinRun, Procgen, and Meta Arcade) necessarily only randomize certain parts of each game. While appropriate for research in generalization, this scenario differs from the typical multi-task learning setting which generally includes more significant differences between tasks. Finally, some benchmarks are affected by performance or usability issues. In particular, MineRL suffers from the relatively high computational cost of the underlying game and gym-retro requires substantial task-specific preprocessing as mentioned above.

3 Dojo benchmark specification

The main goal of Dojo is to provide a highly challenging environment that is only efficiently solvable by generalizing and transferring knowledge across games. To this end, the benchmark includes many similar tasks to provide ample opportunities for positive knowledge transfer as well as many dissimilar tasks to evaluate how well agents can avoid negative knowledge transfer and task interference problems. Finally, the benchmark is easy to use with existing algorithms and includes all necessary tools for preprocessing and evaluation.

3.1 Tasks

The benchmark consists of around 1100 video games from which almost 2000 different RL tasks can be derived. These games include 104 games drawn from Atari (via the ALE), 925 games from the SNES, NES, GameBoy, Genesis and SMS game consoles (via the gym-retro library), 16 games from the Procgen library, and 23 games that are adaptations of those included in the Meta Arcade library. For many of these games, several variants or options are available: all procgen and MetaArcade games support customizing their graphical appearance and many gym-retro games provide several different levels and difficulty options. By exposing each game variant as a separate task to be solved, the variety within the benchmark is increased further and brings the total number of tasks close to 2000. A small selection of games from the benchmark can be seen in Figure 1.

In an effort to standardize the use of Dojo we defined several subsets of tasks intended for different kinds of experiments. These are supplied in the `dojo.tasks` module. In addition to the full

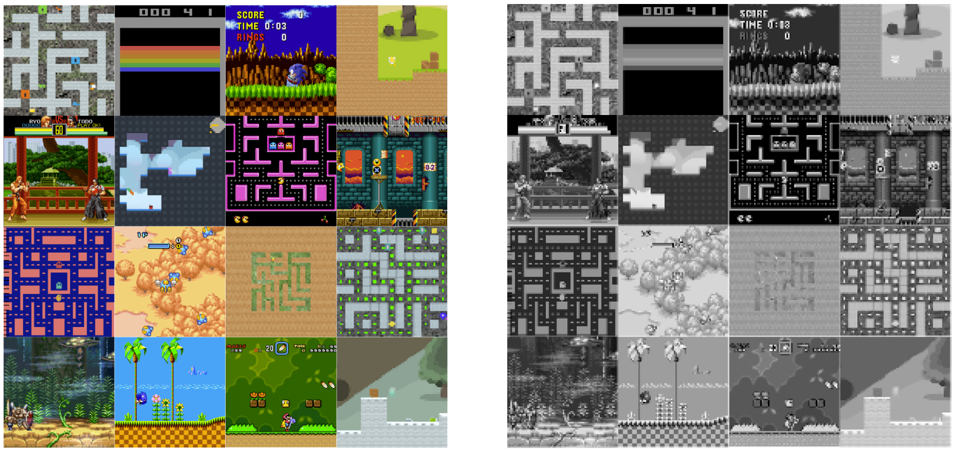


Figure 1: Sample of 16 games from Dojo in native resolution and full color (left) and after default preprocessing steps are applied (right).

set of tasks (`dojo_full`), we provide a slightly reduced version (`dojo_core`), which excludes 350 hard-exploration tasks with exceptionally sparse rewards. We expect work focusing on MTRL, generalization, and transfer learning to primarily use the core set while work on exploration in multi-task settings to use the full benchmark. Additionally, a number of clusters of semantically related tasks (platforming, fighting, and air combat games) are provided as easier test-beds for multi-task learning.

3.2 Actions and observations

To enable simple use of the environment with existing algorithms, action and observation spaces should be shared across all tasks. This is non-trivial, since many of the games differ in the resolution of observations or the kinds of available actions. In particular, games intended for human players were originally developed for different game consoles with different physical controllers.

Dojo therefore uses a unified action space with 15 discrete actions which are mapped to task-specific action schemes in a sensible way. This mapping was chosen such that the semantics of each action are roughly the same across tasks. Actions that are invalid for a certain task are automatically mapped to the "NOOP" action. The exact mapping is provided in Appendix C. For more advanced use, an action mask is provided that can be used with algorithm-specific implementations of invalid action masking such as for instance discussed in Huang & Ontañón (2020). The included Rainbow-DQN example agent implements this by discarding invalid actions both during environment interaction and within the DQN update step. The observations for all tasks consist of the screen content of the respective game, but can differ both in resolution and used color space. Therefore, all observations are converted to grayscale and scaled to a common resolution of 72 × 72 pixels using area interpolation from OpenCV. These hyperparameters were found to be good choices in experiments with a subset of games (see Section 4.3), but can be scaled up depending on the available hardware.

3.3 Rewards

Next we turn our attention to the reward function. Dojo reuses the task-specific reward functions provided by the underlying libraries but optionally applies a normalization step and reward clipping. This normalization is necessary since the unnormalized episodic returns range across ten orders of magnitude as can be seen in Figure 2. Still, use of more advanced normalization schemes such as PopArt is recommended since the applied normalization (dividing by the mean absolute return under the uniform random policy) does not account for the shifting return distribution over the course of training.

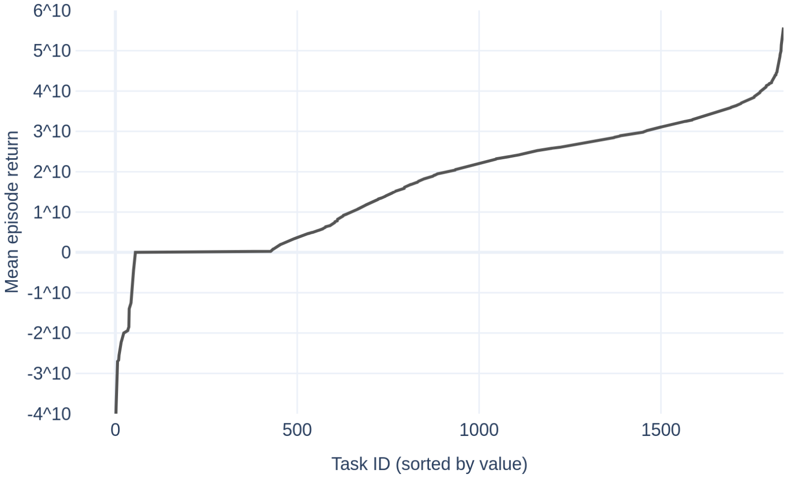


Figure 2: Mean episodic return achieved under the uniform random policy for each of the 2000 tasks.

3.4 Other considerations

Task scheduling. In existing work in MTRL, all tasks to be learned are commonly run in parallel (Espeholt et al., 2018; Hessel et al., 2019). This approach is computationally efficient and also minimizes the issue of catastrophic forgetting compared to executing tasks sequentially. However, due to the large number of tasks and the fact that gym-retro is not thread-safe (meaning each task needs to be run in a separate process), the high resulting memory requirement make this approach infeasible for Dojo. Instead, a pool of worker processes is maintained, each of which executes a single task that is assigned to it by a central task scheduler. Upon episode termination, the scheduler selects a new task to assign to the worker. The default scheduling policy chooses tasks based on the total number of environment interactions allocated to each task so far. To reduce the scheduling overhead, tasks are only replaced after having been run for a minimum of 500 frames in the current session. By changing the order and frequency with which tasks are run, the task scheduler can also induce a specific curriculum on the learning process. Thus, Dojo may also be a useful tool in the area of curriculum learning research.

Seeding. To ensure that results are reproducible, Dojo implements a seeding system that ensures that runs that are initialized with the same seed are exactly equal. In particular, the task scheduling and game initializations are deterministically derived from the given seed.

Statistics & logging. Dojo includes a logging module that continuously records statistics related to the performance of the agent as well as data on internals of the benchmark itself. Data includes per-episode statistics (return, clipped return, episode length, and the episode action distribution) and global statistics (distribution of actions over time, system resource utilization and performance statistics, and statistics related to the task scheduling). Additionally, 30-second video snapshots of all currently active tasks are periodically recorded.

Implementation. Dojo is implemented in Python and makes use of numpy, OpenCV, and the underlying libraries of the individual games. The environment parallelization is done via Ray, which exhibits a similar performance as other implementations of vectorized environments, but allows for simpler dynamic scheduling of tasks.

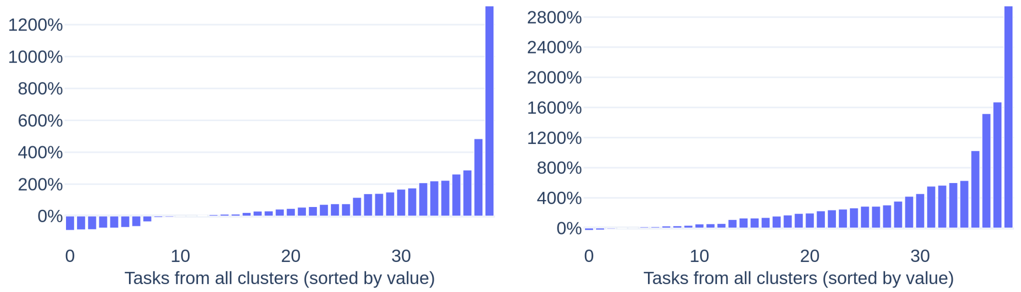
4 Experiments

We perform a number of experiments to (1) determine whether knowledge transfer between tasks in the benchmark genuinely accelerates learning, (2) create a baseline multi-task agent, and (3) find a good choice of preprocessing methods.

4.1 Knowledge Transfer

To evaluate the impact of positive knowledge transfer, we perform the following experiment. We manually determine clusters of semantically related tasks including a cluster of "Pacman"-like games, a cluster of boxing games, and a cluster of air combat games (see Appendix B for the full list). While games within a cluster differ substantially in their visual presentation and precise environment dynamics, they all share some underlying structure and require similar skills. Thus, we expect an agent to perform better at a task when trained jointly with tasks from the same cluster than with unrelated other tasks. To confirm this, we train a Rainbow-DQN (Hessel et al., 2018) agent as implemented in Schmidt & Schmied (2021) on tasks all sampled either from the same cluster or from the union of all clusters. The agent used the IMPALA-CNN architecture from Espeholt et al. (2018) with twice the number of channels and was trained for 20 million frames in each run.

The results presented in Figure 3 show a strong performance improvement in 73% of tasks when trained jointly with tasks drawn from the same cluster. For reference, the mean and median difference in evaluation scores across tasks were 72% and 43% respectively. However, since the maximal achievable scores depend highly on the specific task, the previous metric (percentage of tasks in which the performance improved) is more meaningful. For additional context, the performance difference between the same-cluster trained agent and the uniform random policy baseline, is also provided. Overall, we can conclude that many tasks within Dojo do indeed share some underlying structure making multi-task learning likely a fruitful approach to solving the benchmark.



(a) Per-task performance difference when training on a task together with tasks from the same semantic cluster instead of with semantically unrelated tasks. For example, the agent performed around 1300% better in the rightmost task (`retro:ArtOfFighting-Snes`), when trained jointly with other fighting games, compared to when trained with arbitrary other tasks.

(b) Per-task performance increase of the agent trained on tasks from the same cluster jointly compared to the uniform random policy baseline. While many of the tasks do present a difficult challenge (particularly exploration-wise due to sparse rewards), we can see that the agent does make substantial learning progress even within only 20M frames.

Figure 3: Performance difference between two agents for each of the selected tasks.

4.2 Multi-task learning

Since we were able to confirm the effectiveness of knowledge transfer within small clusters of tasks, we now turn our attention to the full benchmark. To facilitate future research with Dojo, we want to establish a baseline for all tasks and verify that Dojo indeed presents a difficult challenge to current methods. To this end, we train a Rainbow-DQN agent with the IMPALA architecture on the full suite of tasks. Due to compute limitations the agent was trained for only 250 million frames, corresponding to a relatively low 2.5 hours of game play time per task.

While the mean clipped improvement across tasks (clipped at $\pm 200\%$ for each task) was 46%, Figure 4 clearly shows that the agent actually performed worse in a substantial share of tasks and in several tasks the performance fully collapsed to zero. We hypothesize that this is due to the effects of negative knowledge transfer combined with the short training duration. We also identified a further problem whereby the issue of differences in the return scales of different tasks (as discussed in Section 2.1) may be magnified when using prioritized experience replay (PER) (Schaul et al., 2016). This is because PER prioritizes samples from the replay buffer based on the TD-error, potentially leading to a disproportionate oversampling of tasks with large return scales in addition to the imbalance within the update steps themselves.

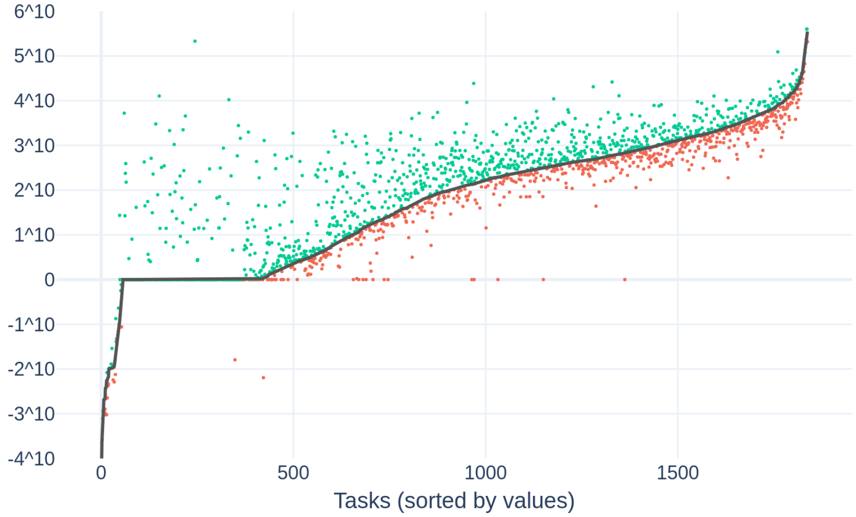


Figure 4: Per-task evaluation scores for the uniform random policy baseline (in black) and the trained agent (in green or red depending on whether it performed better or worse).

4.3 Preprocessing settings

Using the raw high-resolution full-color images as observations preserves the most information, but results in a large memory and compute cost, particularly when using experience replay. Thus, we perform a number of experiments to determine sensible default hyperparameters that maximize resource-efficiency while not impeding learning progress due to too much loss of information. We train a Rainbow-DQN agent individually on 12 randomly chosen tasks for 3.2M frames each, using the parameters listed in Table 1.

Table 1: Each row of hyperparameters was evaluated in 12 different tasks and averaged across 2 random seeds.

Grayscale	Resolution
No	72 × 72
Yes	72 × 72
Yes	64 × 64
Yes	72 × 72
Yes	80 × 80
Yes	88 × 88

Among color and grayscale observations there was no consistently better option, indicating that color information is largely irrelevant for solving these tasks. Overall, we found grayscale observations with a resolution of 72 × 72 pixels to strike a good balance between performance and resource-efficiency. With this setting, an efficiently implemented DQN agent would require approximately 5GB of memory for its replay buffer. The full results can be found in Appendix A.

5 Conclusion and future work

In this work, we provided an overview of current methods and open problems in multi-task reinforcement learning, discussed a number of shortcomings affecting existing benchmarks in this setting, and introduced a new benchmark intended to support future research in this area. We performed an array of experiments that aided in the design of the benchmark, helped to better understand the role of knowledge transfer in small task clusters, and yielded a preliminary baseline for the benchmark as a whole.

In the future we intend to implement several additional improvements to the benchmark. These include enhancements to the invalid action masking to allow for more fine-grained masking for Atari games, implementation of other useful task-schedulers, and the option to use custom schedulers for better usability in curriculum learning research. Using these tools, we then plan to perform a more in-depth analysis of existing multi-task learning methods, particularly in regard to how they can be used to improve data-efficiency and speed up exploration.

REFERENCES

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020.

Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. Deepmind lab. abs/1612.03801, 2016. URL `http://arxiv.org/abs/1612.03801`.

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*, 47:253–279, Jun 2013.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Karl Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1282–1289. PMLR, 2019.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*. PMLR, 2018.

William Fedus, Dibya Ghosh, John D. Martin, Marc G. Bellemare, Yoshua Bengio, and Hugo Larochelle. On catastrophic interference in atari 2600 games. abs/2002.12499, 2020. URL `https://arxiv.org/abs/2002.12499`.

William H. Guss, Mario Ynocente Castro*, Sam Devlin*, Brandon Houghton*, Noboru Sean Kuno*, Crissman Loomis*, Stephanie Milani*, Sharada Mohanty*, Keisuke Nakata*, Ruslan Salakhutdinov*, John Schulman*, Shinya Shiroshita*, Nicholay Topin*, Avinash Ummadisingu*, and Oriol Vinyals*. Neurips 2020 competition: The MineRL competition on sample efficient reinforcement learning using human priors.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pp. 3215–3222, 2018.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 3796–3803. AAAI Press, 2019.

Jacob Hilton, Nick Cammarata, Shan Carter, Gabriel Goh, and Chris Olah. Understanding rl vision. *Distill*, 2020. doi: 10.23915/distill.00029. https://distill.pub/2020/understanding-rl-vision.

Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. abs/2006.14171, 2020. URL `https://arxiv.org/abs/2006.14171`.

Steven Kapturowski, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning, 2021.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. *ArXiv*, abs/2004.14990, 2020.

Michael Luo, Jiahao Yao, Richard Liaw, Eric Liang, and Ion Stoica. IMPACT: importance weighted asynchronous architectures with clipped target networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *J. Artif. Intell. Res.*, 61:523–562, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.

Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. Gotta learn fast: A new benchmark for generalization in RL. abs/1804.03720, 2018. URL `http://arxiv.org/abs/1804.03720`.

Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *ArXiv*, abs/2006.12862, 2020.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*, 2016.

Tom Schaul, Georg Ostrovski, Iurii Kemaev, and Diana Borsa. Return-based scaling: Yet another normalisation trick for deep RL. abs/2105.05347, 2021. URL `https://arxiv.org/abs/2105.05347`.

Dominik Schmidt and Thomas Schmied. Fast and data-efficient training of rainbow: an experimental study on atari. abs/2111.10247, 2021. URL `https://arxiv.org/abs/2111.10247`.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, and et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, Dec 2020.

Rohin Shah, Cody Wild, Steven H. Wang, Neel Alex, Brandon Houghton, William Guss, Sharada Mohanty, Anssi Kanervisto, Stephanie Milani, Nicholay Topin, Pieter Abbeel, Stuart Russell, and Anca Dragan. NeurIPS 2021 competition proposal: The MineRL BASALT competition on learning from human feedback. *NeurIPS Competition Track*, 2021.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Edward W. Staley, Chace Ashcraft, Benjamin Stoler, Jared Markowitz, Gautam Vallabha, Christopher Ratto, and Kapil D. Katyal. Meta arcade: A configurable environment suite for meta-learning. abs/2112.00583, 2021. URL `https://arxiv.org/abs/2112.00583`.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michaël Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents. abs/2107.12808, 2021. URL `https://arxiv.org/abs/2107.12808`.

Hado van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. Learning values across many orders of magnitude. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4287–4295, 2016.

Nelson Vithayathil Varghese and Qusay H. Mahmoud. A survey of multi-task deep reinforcement learning. *Electronics*, 9(9), 2020.

A Preprocessing experiments

A.1 Color space

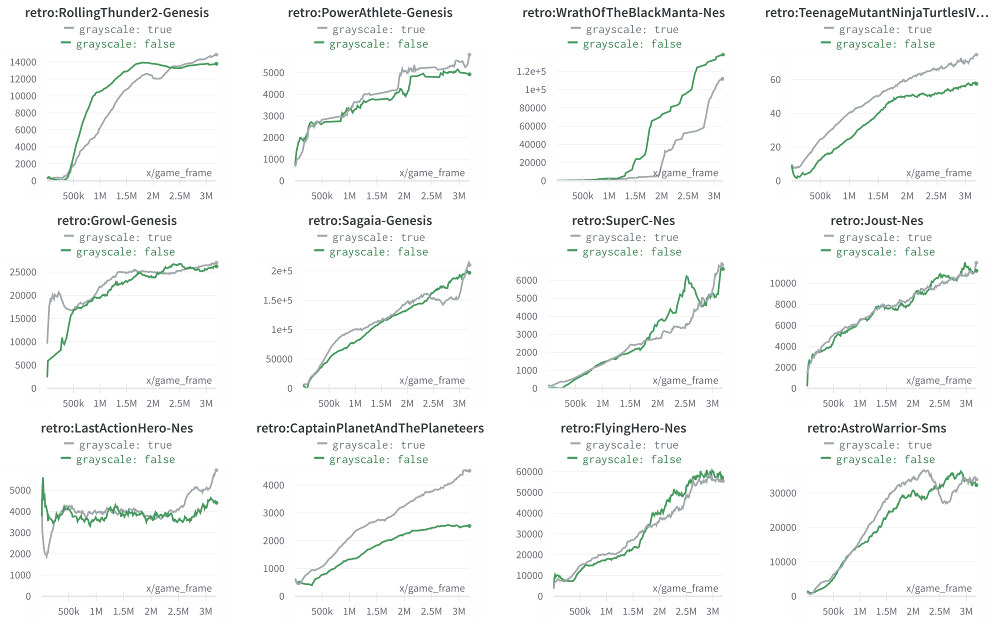


Figure 5: Learning curves for each of the twelve tasks when using grayscale or full-color observations.

A.2 Resolution

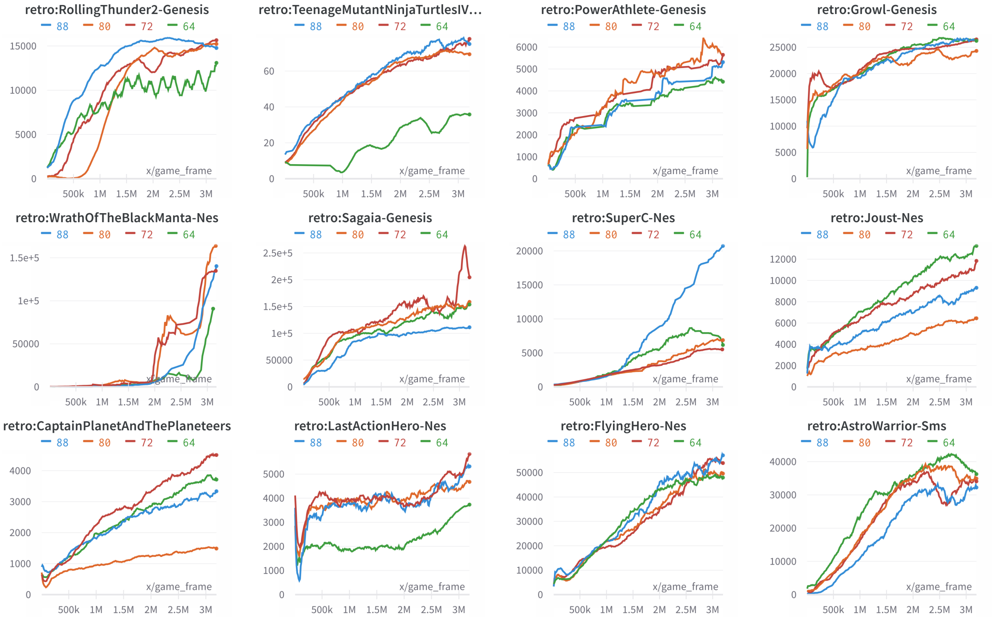


Figure 6: Learning curves for each of the twelve tasks when using the specified resolutions for the observations.

B Task clusters

Table 2: Clusters of tasks used in knowledge transfer experiments.

Cluster name	**Games**
pacman	retro:MsPacMan-Nes
	atari:ALE/MsPacman-v5
	procgen:chaser
	atari:ALE/Pacman-v5
	retro:PacMania-Genesis
	retro:MsPacMan-Genesis
	retro:MsPacMan-Sms
	retro:PacManNamco-Nes
	retro:PacMania-Sms
space-invaders	retro:SuperSpaceInvaders-Sms
	retro:SpaceInvaders-Nes
	retro:SpaceInvaders91-Genesis
	retro:SpaceInvaders-Snes
	atari:ALE/SpaceInvaders-v5
breakout	retro:BlockKuzushiGB-GameBoy
	atari:ALE/Breakout-v5
	retro:BlockKuzushi-Snes
	retro:Alleyway-GameBoy
paperboy	retro:Paperboy-Sms
	retro:Paperboy2-Genesis
	retro:Paperboy-Nes
	retro:Paperboy-Genesis
air-combat	retro:EarthDefenseForce-Snes
	retro:OverHorizon-Nes
	retro:Hellfire-Genesis
	retro:Gradius-Nes
	retro:Sagaia-Genesis
	retro:ZeroWing-Genesis
pong	meta_arcade:meta_arcade_levels/cst_pong.json
	meta_arcade:meta_arcade_levels/cst_pong_breakout.json
	meta_arcade:meta_arcade_levels/cst_battle_pong.json
	atari:ALE/Pong-v5
boxing-fighting	retro:FinalFight2-Snes
	retro:ArtOfFighting-Snes
	retro:PitFighter-Sms
	retro:FinalFight-Snes
	retro:ArtOfFighting-Genesis
	retro:FinalFightGuy-Snes
	retro:TeenageMutantNinjaTurtlesTournamentFighters-Nes
	retro:FinalFight3-Snes
	retro:PitFighter-Genesis
	retro:TeenageMutantNinjaTurtlesTournamentFighters-Genesis

C Action mapping

Table 3: The 15 available actions are mapped to the specified task-type-specific actions in the following way. This choice of mapping ensures that the effect of each action is relatively similar across all tasks.

#	**Dojo action**	**atari**	**procgen**	**NES**	**SNES**	**Genesis**	**SMS**	**GameBoy**
0	NOOP	NOOP	()	None	/	/	None	None
1	FIRE	FIRE	D	B	Y	B	B	B
2	UP	UP	UP	UP	UP	UP	UP	UP
3	RIGHT	RIGHT	RIGHT	RIGHT	RIGHT	RIGHT	RIGHT	RIGHT
4	LEFT	LEFT	LEFT	LEFT	LEFT	LEFT	LEFT	LEFT
5	DOWN	DOWN	DOWN	DOWN	DOWN	DOWN	DOWN	DOWN
6	UP_RIGHT	UPRIGHT	RIGHT+UP	UP+RIGHT	UP+RIGHT	UP+RIGHT	UP+RIGHT	UP+RIGHT
7	UP_LEFT	UPLEFT	LEFT+UP	UP+LEFT	UP+LEFT	UP+LEFT	UP+LEFT	UP+LEFT
8	DOWN_RIGHT	DOWNRIGHT	RIGHT+DOWN	DOWN+RIGHT	DOWN+RIGHT	DOWN+RIGHT	DOWN+RIGHT	DOWN+RIGHT
9	DOWN_LEFT	DOWNLEFT	LEFT+DOWN	DOWN+LEFT	DOWN+LEFT	DOWN+LEFT	DOWN+LEFT	DOWN+LEFT
10	JUMP	UPFIRE	A	A (jump)	B	C	A	A
11	KICK	DOWNFIRE	W	A+B (kick)	A	DOWN+B		UP+DOWN
12	SPEC0	RIGHTFIRE	S	UP+B (special)	X	A		DOWN+A
13	SPEC1	LEFTFIRE	Q	DOWN+A (squat)	L	Z		
14	SPEC2		E		R	X		