Expert-augmented actor-critic for ViZDoom and Montezuma's Revenge

Michał Garmulewicz
University of Warsaw

Henryk Michalewski, Piotr Miłoś
University of Warsaw
deepsense.ai

Abstract

We propose an expert-augmented actor-critic algorithm, which we evaluate on two environments with sparse rewards: Montezuma's Revenge and a demanding maze from the ViZDoom suite. In the case of Montezuma's Revenge, an agent trained with our method achieves very good results consistently scoring above 27,000 points (in many experiments beating the first world). With an appropriate choice of hyperparameters, our algorithm surpasses the performance of the expert data. In a number of experiments, we have observed an unreported bug in Montezuma's Revenge which allowed the agent to score more than 800, 000 points.

1 Introduction

Deep reinforcement learning has shown impressive results in simulated environments (MKS$^+$15; MBM$^+$). However, current approaches often fail when rewards are sparse as the cost of random exploration increases rapidly with the distance to rewards. This substantially restricts the scope of possible real-world applications, e.g. rewards in robotics are often assigned only once a task is completed and thus are binary and sparse (AWR$^+$17; VHS$^+$). Additionally, random exploration potentially leads to safety issues.

One way to improve the efficiency of exploration is to utilize expert data. The simplest technique, behavioral cloning, often suffers from compounding errors when drifting away from supervisor's demonstrations, see also Figure 7 in Section 6. This can be mitigated by iterative methods like DAgger (RGB) at the cost of a cumbersome data collection process. Recent work (HVP$^+$) has analyzed performance of behavioral cloning on Atari 2600 games. In the case of Montezuma's Revenge, cloning of near-perfect human demonstrations (a score above 30,000 points) led to a policy which scores merely 575 points.

We aim to address the shortcomings of cloning methods by combining the ACKTR algorithm (WML$^+$17) with supervised learning from the expert data. Our method is easy to understand and implement. The core of our contribution can be expressed by a single formula, see Equation (1). The expert data is intended to guide the agent's exploration. We evaluate our method on two environments with sparse rewards: Montezuma's Revenge and a demanding MyWayHome maze from the ViZDoom suite [1]. Interestingly, our method is able to achieve stronger performance than the expert data, see Figure 3.

We describe our method in Section 3. In Section 4 we describe environments, training procedures, expert data and neural architectures. In Section 5 we summarize our experiments. Section 6 contains conclusions and suggestions of further work. Code for experiments presented in this paper is publicly available: `https://github.com/ghostFaceKillah/expert`.

2 Related work

Deep reinforcement learning The celebrated paper (MKS$^+$15) has sparked rapid development of deep reinforcement learning. The DQN algorithm presented there for the first time achieved (super)human performance in many Atari games. Soon after, other algorithms were developed e.g. A3C (MBM$^+$) and TRPO (SLA$^+$15). In this paper we employ the Actor-Critic using Kronecker-Factored Trust Region (ACKTR) algorithm (WML$^+$17). ACKTR is a policy-gradient method that utilizes natural gradient techniques (Ama98) which have proved to be successful in increasing speed and stability of learning.

Expert data in reinforcement learning. A vast body of literature has been devoted to imitation learning. As the learned policy influences the future states upon which it will be evaluated, errors compound and make this method impractical for tasks with long time horizon (RB) (e.g. behavioral cloning in Montezuma's Revenge (HVP$^+$) yielded lackluster performance). The seminal work (RGB) describes the DAgger algorithm that alleviates this issue, however at the cost of the iterative expert data collection process. In our method, we collect a fixed set of expert data, which is reused in the whole experiment.

Value-based deep reinforcement learning methods have been combined with learning from demonstrations (HVP$^+$; PPH$^+$18). In these works, the learning algorithm is altered to ensure that the expert data added to the replay buffer is used efficiently. For example, no action is allowed to have a higher

[1] Please see videos of Montezuma's Revenge gameplay (`https://bit.ly/2wG8Zh5`), exploiting an unreported bug in Montezuma's Revenge (`https://bit.ly/2PC7cRi`) and ViZDoom MyWayHome gameplay (`https://bit.ly/2wMlxDL`)

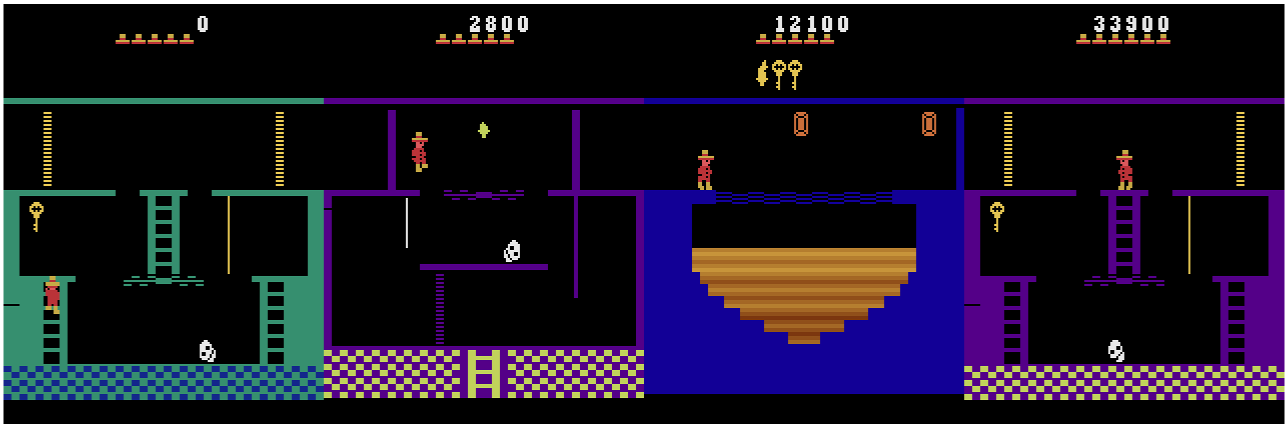


Figure 1: Montezuma's Revenge - four screenshots from various stages of a gameplay. The room on the right marks the beginning of the second world. The agent reaches this room after passing through all the rooms in the first world. For the map of the first world see `https://atariage.com/2600/archives/strategy_MontezumasRevenge_Level1.html`*, for the map of the second world see* `https://atariage.com/2600/archives/strategy_MontezumasRevenge_Level2.html`*.*

Approach	**Mean score**	**Max score**	**Trans. $\times 10^6$**	**Methods used**
ExpAugAC (our method)	27,052	804,900	200	Expert loss term based on batches of prerecorded data
DQfD (HVP$^+$)	4,740	-	200	750k batches of expert pretraining, 3 additional loss terms, prioritizing expert data replay
Behavioral cloning (HVP$^+$)	575	-	24	–
Ape-X DQfD (PPH$^+$18)	29,384	-	2,500	Methods from DQfD, temporal consistency loss, transformed Bellman operator
Playing hard YT (APB$^+$18)	41,098	-	1,000	Auxiliary reward encouraging imitation of videos of expert gameplays
Learning MR from a Single Demonstration (SC18)	-	74,500	-	Decomposing task into a curriculum of shorter subtasks, assumes ability to set env. state
Unifying Count-Based Exploration (BSO$^+$16)	3,439	6,600	100	Exploration-based auxiliary reward, mixing Double Q-learning target with MC return

Table 1: The state of the art for Montezuma's Revenge.

Q-value than expert's action, the n-step return loss is used to propagate expert information and expert demonstrations are sampled more often. A different approach, combining DQN with expert data classification loss has been presented in (LOB). However, authors do not test their method on Montezuma's Revenge or ViZDoom. In our work, we only use one expert imitation loss. Formulation of this loss is motivated by the policy gradient theorem, see Formula (1).

Methods combining policy gradient algorithm with expert demonstrations have been successfully applied to problems such as complex robotic manipulation (RKG$^+$17) or dialogue management (SBU$^+$17). In the case of Atari environments, in (ZM18) expert data is used to pre-train policies that are later passed to a policy gradient algorithm. A substantial effort has been put into integrating replay buffers into actor-critic algorithms including (Waw09; WT13; WBH$^+$).

Montezuma's Revenge Significant research attention has been dedicated to Montezuma's Revenge as it is regarded as a testing ground that challenges exploration abilities of a given learning algorithm. Table 1 summarizes the current state of the art, including methods using expert data. Efforts have been made to tackle this environment by adding natural language instructions (KSS17) (avg. 3,500 pts using 60M transitions), extending model with intrinsic curiosity rewards in (OBvdOM17) (avg. 3,700 pts using 150M transitions) or introducing hierarchy into the model (VOS$^+$) (avg. 2600 pts using 1000M frames). Numbers in the parenthesis represent scores and amount of training samples reported in these papers. Combining hierarchical learning and imitation learning has been demonstrated to be effective (LJA$^+$18), although the work focuses only on the first screen of Montezuma's Revenge.

Doom ViZDoom (KWR$^+$) is a popular suite of reinforcement learning environments based on the first-person shooter game Doom. Impressive results have been obtained for various tasks in these environments. These tasks include in particular navigation (SDK18; LPCS; PS17) and multi-player combat (DK16; CL17; WT; KSGG16). In (ADKB18; JLK17; DSC$^+$16; CLSS16) the ViZDoom suite was used as a benchmark in other reinforcement learning experiments.

Curiosity-based intrinsic rewards have been used in (PAED17) to increase efficiency of exploration in ViZDoom. Authors of (PAED17) report that in the navigation task MyWayHome (see Section 4 for a description) the standard actor-critic approach completely fails, while their method solves the task in 70% of the cases after using 10M transitions for training. Our algorithm always solves the task after training on 5M transitions. To the best of our knowledge, our method is the first one to use expert data in ViZDoom.

3 Expert-augmented ACKTR

Reinforcement learning is formalized in the framework of Markov decision processes (MDP). In this setting at each timestep t, the agent observes a state $s_t \in S$ and chooses an action $a_t \in A$ according to policy π, which is a (potentially stochastic) mapping from a given state to actions.

The environment then returns reward $r(s_t, a_t)$ and evolves to the next state s_{t+1} according to a transition probability $P(s_{t+1}|s_t, a_t)$. The goal of reinforcement learning is to find a policy within some class which maximizes

$$\mathcal{J}(\pi) = \mathbb{E}_\pi[R_t],$$

where

$$R_t = \sum_{i=0}^{\infty} \gamma^i r(s_{t+i}, a_{t+i}),\ \gamma \in (0,1),$$

is the future discounted reward. In our applications the set of available neural network policies is parametrized by θ. Policy-based methods (SML$^+$15; SMSM99; Wil92) are inspired by the Policy Gradient Theorem which states that

$$\nabla_\theta \mathcal{J}(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_t \mathrm{adv}_t \nabla_\theta \left(\log \pi_\theta(a_t|s_t)\right)\right],$$

where

$$\mathrm{adv}_t = R_t - b(s_t)$$

and b is an arbitrary function of the state, called a *baseline* function. In the context of A2C and ACKTR algorithms we take the approximate state-value function V_θ as a baseline. Thus we aim to optimize the following loss

$$L^{\mathrm{A2C}}(\pi_\theta) = \sum_t L_t^{\mathrm{A2C}}(\pi_\theta),$$

where

$$L_t^{\mathrm{A2C}}(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[-\mathrm{adv}_t \log \pi_\theta(a_t|s_t) + \frac{1}{2}(R_t - V_\theta(s_t))^2\right].$$

The main algorithmic contribution of our paper consists in adding an extra term to the loss function to accommodate the expert data:

$$L(\pi_\theta) = L^{\mathrm{A2C}}(\pi_\theta) - \frac{\lambda_{\mathrm{expert}}}{k} \sum_{i=1}^{k} \mathrm{adv}_i^{\mathrm{expert}} \log \pi_\theta(a_i^{\mathrm{expert}}|s_i^{\mathrm{expert}}), \tag{1}$$

where a_i^{expert} and s_i^{expert} are sampled from a fixed dataset of expert rollouts.

We consider three variants of the expert advantage:

$$\textit{reward: } \mathrm{adv}_t^{\mathrm{expert}} = \sum_{s\geq 0} \gamma^s r_{t+s}^{\mathrm{expert}},$$

$$\textit{critic: } \mathrm{adv}_t^{\mathrm{expert}} = \left[\sum_{s\geq 0} \gamma^s r_{t+s}^{\mathrm{expert}} - V_\theta(s_t)\right]_+,$$

$$\textit{simple: } \mathrm{adv}_t^{\mathrm{expert}} = 1.$$

where $[x]_+ = \max(x, 0)$. We follow the ACKTR algorithm which estimates the natural gradient direction associated with the current policy, see Algorithm 1 for additional details.

4 Environments, training and architectures

Doom MyWayHome is a navigation task defined in the ViZDoom suite (KWR$^+$). In MyWayHome the maze consists of 9 rooms, see Figure 5.

The agent receives reward 1 if it reaches the *goal* location in less than 2100 time steps. There are not other rewards in the game. The *goal* is always located in the same place and the agent always starts in a fixed location far from the *goal*. The action space consists of three actions: *forward*, *turn left*, *turn right*.

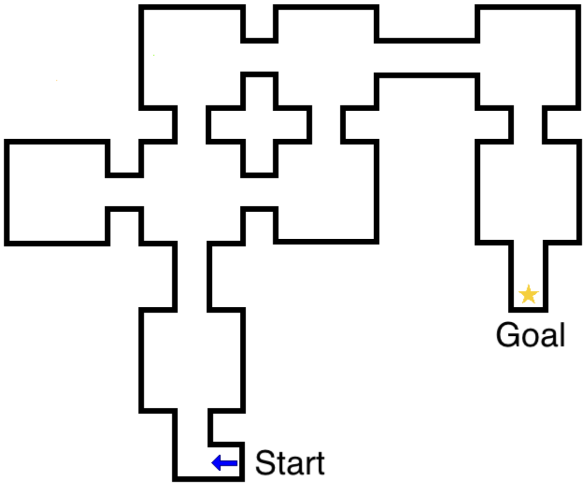


Figure 5: The map of the MyWayHome maze. The starting position and the goal are always located in the same place. See Figure 6 for a screenshot from the environment.

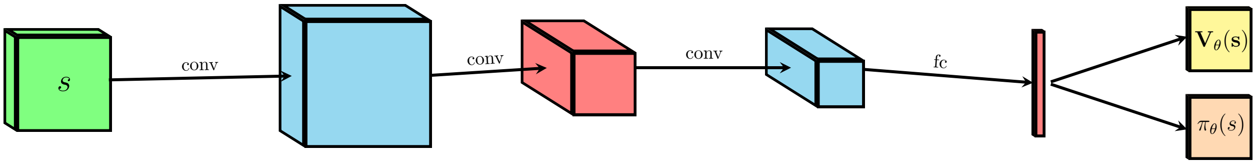


Figure 2: Network architecture: stacked input frames are passed through a convolution layer with 32 filters size of 8 and stride 4, followed by 64 convolution filters of size 4 and stride 2, followed by 64 convolution filters of size 3 and stride 1. The output is fed into a fully connected layer with 512 units. Next, two heads are attached. The first one with a single neuron for the value function approximation and the second one with the number of neurons matching the dimension of the action space. After applying the softmax function the second head outputs the policy. In all layers, except the top heads, we use the ReLU non-linearity.

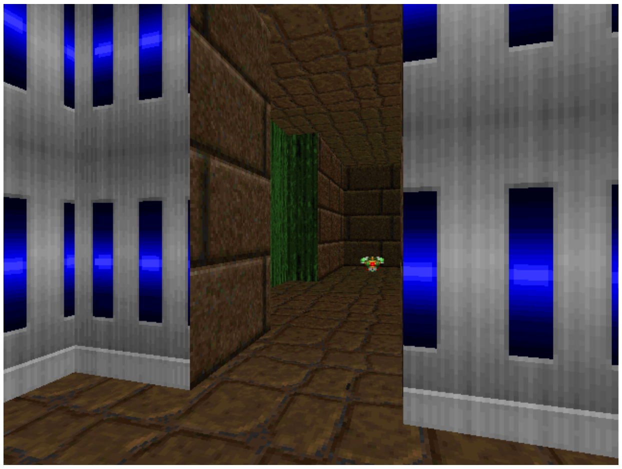


Figure 6: An observation from the MyWayHome environment. The goal location is visible in the background in the next room.

Montezuma's Revenge Montezuma's Revenge belongs to the suite of Atari 2600 games (BNVB13). The task in the game is to navigate in a labyrinth (see Figure 1), collect valuable items and avoid deadly obstacles. Montezuma's Revenge is known for its sparse rewards. For example, obtaining the only reward available in the first room (see the leftmost image in Figure 1) requires five "macro actions" such as "jump across a gap" or "go down a ladder", each spanning over dozens of environment steps.

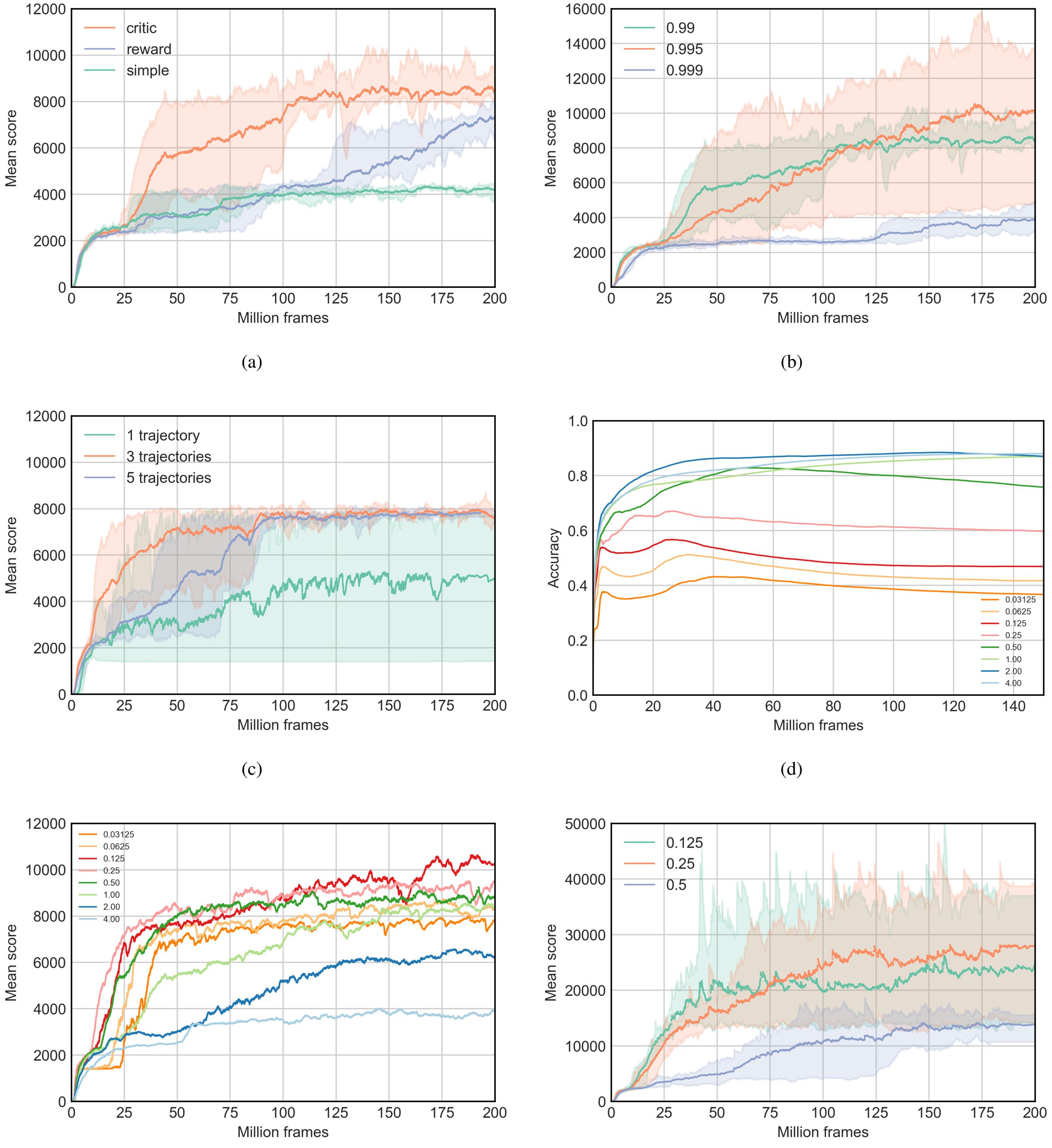
Training details Training details are shared across Montezuma's Revenge and ViZDoom's MyWayHome environments. The input to the policy consists of the stacked last four frames which are downscaled to 84 × 84 pixels and cast to grayscale. We use Algorithm 1 with the learning rate 0.125, 32 actors and the time horizon $T = 20$.

For the expert data we use $k = 256$ (see Equation 1). Following (MBM$^+$), given entropy H we introduce an additional entropy regularization term: β_{entropy} defined as $\nabla_\theta H(\pi_\theta(\cdot|s_t))$. We set $\beta_{\text{entropy}} = 0.001$.

Our implementation of Algorithm 1 is based on the implementation of the ACKTR algorithm provided in (DHK$^+$17).

```
Data: Dataset of expert transitions
      $(s_t^{\text{expert}}, a_t^{\text{expert}}, s_{t+1}^{\text{expert}}, r_t^{\text{expert}})$
Initialize neural net policy $\pi_\theta$
for $iteration \leftarrow 1$ to $max\ steps$ do
    for $t \leftarrow 1$ to $T$ do
        Perform action $a_t$ according to $\pi_\theta(a|s_t)$
        Receive reward $r_t$ and new state $s_{t+1}$
    end
    for $t \leftarrow 1$ to $T$ do
        Compute discounted future reward
          estimator: $\hat{R}_t = r_t + \gamma r_{t+1} + \ldots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V_\theta(s_T)$
        Compute advantage estimator:
          $\widehat{\text{adv}}_t = \hat{R}_t - V_\theta(s_t)$
    end
    Compute $\nabla$ of the A2C loss $g_{A2C} = \nabla_\theta \frac{1}{T} \sum_{t=1}^{T} \left[ -\widehat{\text{adv}}_t \log \pi_\theta(a_t|s_t) + \frac{1}{2}(\hat{R}_t - V_\theta(s_t))^2 \right]$
    Sample mini-batch of $k$ expert state-action
      pairs
    Compute expert advantage estimator
      $\text{adv}_t^{\text{expert}}$ for each state-action pair
    Compute $\nabla$ of the expert loss $g_{\text{expert}} = \nabla_\theta \sum_{i=1}^{k} \left[ -\text{adv}_i^{\text{expert}} \log \pi_\theta(a_i^{\text{expert}}|s_i^{\text{expert}}) \right]$
    Update ACKTR inverse Fisher estimate
    Use ACKTR Kronecker optimizer with
      gradient $g = g_{A2C} + \lambda_{\text{expert}} g_{\text{expert}}$.
end
```

Algorithm 1: Expert-augmented ACTKR. Pseudocode for one actor.



(a)

(b)

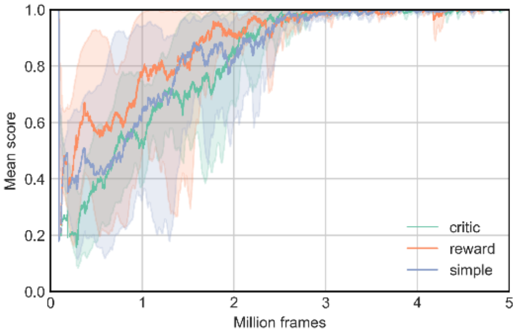
(c)

(d)

(e)

(f)

Figure 3: Montezuma's Revenge experimental results. See the top left plot for a comparison between various advantage estimators defined in Section 3. In experiments depicted in the top right plot we vary the discount rate γ. The sweet spot is around $\gamma = 0.995$. We conjecture that even higher values could be beneficial, but would require better handling of the variance of empirical rewards. Plot in the center-left shows that smaller amount of expert data yields higher variance of results. In the bottom-left plot we test various choices of λ_{expert} to assess utilization of the expert data. The performance peaks for the intermediate values. For small values expert's learning signal is too weak, while high values impair the final performance because the agent overfits to the expert data, as seen in the train set accuracy plot (center-right). In the experiments presented in the bottom-right plot we provide scores for optimized parameters, that is we apply $\gamma = 0.995$, the critic advantage estimator and selected values of λ_{expert}.



M of frames	curriculum	our
1	0.042	0.484
2	0.052	0.790
5	0.040	0.999
10	0.041	0.999
50	0.482	0.999
100	0.949	0.999

Figure 4: Left: In ViZDoom behavior is similar for all expert advantage estimators. Right: performance of a curriculum learning in MyWayHome compared to our expert-augmented algorithm. The curriculum consists in re-spawning the agent in random locations. We experimentally verified that the ACKTR algorithm without the curriculum was unable to solve the MyWayHome task. This echoes observations made for another actor-critic model-free algorithm in (PAED17). Our behavioral cloning experiments also failed to solve this task. The curiosity-based method described in (PAED17) achieves an average score 0.7 *after 10M of frames.*

5 Experiments

In this Section we empirically verify a number of properties of our algorithm. In particular (see Figure 3 for a summary):

1. we check which of proposed advantage estimators offers the best performance,
2. we check how the discount factor γ influences the reward and what are consequences of decreasing the number of expert trajectories,
3. we verify what is the potential for generalization: in the case of Montezuma's Revenge we provided expert trajectories only from the first world of Montezuma's Revenge, but our agents are capable of continuing the gameplay in the second world. For the map of the first world see `https://atariage.com/2600/archives/strategy_MontezumasRevenge_Level1.html`, for the map of the second world see `https://atariage.com/2600/archives/strategy_MontezumasRevenge_Level2.html`
4. we also grid the λ_{expert} coefficient with respect to the total reward and verify how λ_{expert} is related to the accuracy on expert data.

Montezuma's Revenge

For a summary of the state of the art compared with our results see the Table on page 2, for evaluation videos see `https://bit.ly/2wG8Zh5`. Interestingly, our algorithm discovered a bug, see video `https://bit.ly/2PC7cRi`, which manifests through scores exceeding $800,000$ in some evaluation rollouts (these rollouts are excluded when we calculate the mean score). In all experiments reported in this Section we performed at least 5 experiments with different seeds. Shades represent min-max ranges. In all experiments except for Figure 3 top-right and bottom-right, we set $\gamma = 0.99$; in the top-left, top-right and center-left experiments we set $\lambda_{\text{expert}} = 1$; in all experiments except for Figure 3 top-left we use the critic advantage estimator and in all experiments except for Figure 3 center-left we use all available expert trajectories (14 trajectories). In the experiment presented in the bottom-right we selected the best performing parameters from previous experiments, that is the critic advantage estimator and $\gamma = 0.995$. We also set the advantage estimator for expert-data λ_{expert} equal to $0.125, 0.25$ or 0.5.

In order to address the question of generalization from the first Montezuma's Revenge world to the second Montezuma's Revenge world (see the third point at the beginning of this Section), we considered 6 runs for the best parameters ($\lambda_{\text{expert}} = 0.125$, $\gamma = 0.995$). Please note that all provided expert trajectories were limited to the first world. Let us choose a training run which achieves the median score at the end of training. The stochastic policy associated with this run receives average reward $27,052$ in 100 evaluations. It completes the first world in 60% of the cases and outperforms the best expert trajectory in 16% of the cases. In contrast, if we choose the policy not from the median, but from the best run, it resolves the first world in 88% of the cases and outperforms the best expert trajectory in 86% of the cases. Interestingly, in some rollouts (not taken into the average) our policies exploit an unreported bug in Montezuma's Revenge occasionally scoring over $100,000$ points with a maximum of $804,900$ points.

ViZDoom MyWayHome

In Figure 4 we summarize an experiment which compares the efficiency of the model-free ACKTR algorithm with our expert-augmented method. The model-free approach without the expert data does not work. However, the agent learns how to navigate the maze after a relatively mild simplification of the environment consisting in allowing the agent to re-spawn in random locations in the labyrinth, which can be seen as a curriculum learning. The agent manages to learn a consistent behavior in 100M actions. However, if we allow expert data, the agent reaches even a better performance already after 5M actions without any curriculum, see Figure 4.

6 Conclusions and future work

Based on experimental results we claim that the algorithm presented in this work is a practical method of getting good performance in cases when multiple interactions with the environment are possible and good quality expert data is available. It could be particularly useful in settings such as Montezuma's Revenge, where neither supervised learning from expert data nor random exploration yield good results.

Using model-free reinforcement learning algorithms with expert data brings a number of issues which are worth addressing. In particular, our expert-augmented algorithm would benefit from an adaptive λ_{expert} coefficient and accuracy rate targeting on expert trajectories (from our experiments follows that for overall performance of the algorithm it would be most beneficial to keep accuracy between 0.6 and 0.9, see Figure 3). In our view, another interesting research direction would be to design a proxy metric which would identify situations requiring additional expert data. That is situations where we are far away from the expert trajectory and simultaneously the model-free algorithm does not have a capacity to find any rewards on its own, see Figure 7. One can regard this as an automation of the DAgger algorithm (RGB).

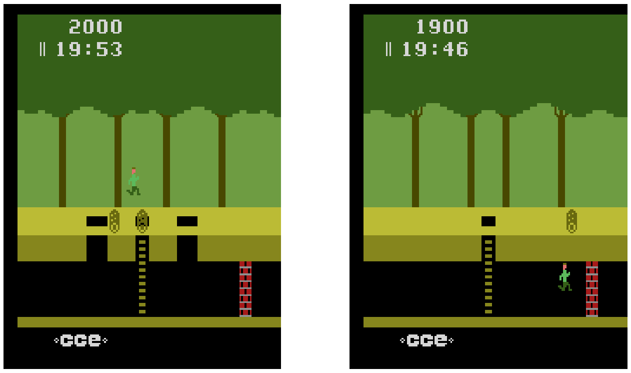


Figure 7: Left: our agent in Pitfall! trained on an expert trajectory which achieves the perfect score (see a run https://youtu.be/MX_e2UZLk1E *of our agent). The expert trajectory and our agent run only above the ground. Right: the agent falls into the lower level and is unable to return to the expert trajectory, see video* https://youtu.be/07327Eh6zIM.

We also leave as future work the following other extensions of the experiments presented in this article:

1. Measure performance of the algorithm when the quality of expert data varies.

2. Rewards in robotics are often assigned only once a task is completed and thus are binary and sparse (AWR+17; VHS+). In our view it would be interesting to test methods presented in this paper on robotic object manipulation tasks or in a simulated car environment.

3. Authors of (PPH+18, Table 3) report Bank Heist, Gravitar, Ms. Pacman, Pitfall! and Solaris as games where human experts are still stronger than AI even when expert trajectories are available in the process of training. In an on-going experiment, we assess the performance of our method in these environments.

4. Using human knowledge in tasks routine for humans but difficult for artificial intelligence, such as driving, seems to be a particularly important application of methods presented in this paper. We test these ideas in an on-going experiment with the Carla simulator (DRC+17).

5. In our experiments, the discount factor $\gamma = 0.995$ performed optimally, see Figure 3. Can we get further to higher values of γ?

References

[ADKB18] Artemij Amiranashvili, Alexey Dosovitskiy, Vladlen Koltun, and Thomas Brox. TD or not TD: Analyzing the role of temporal differencing in deep reinforcement learning. In *ICLR*, 2018.

[Ama98] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[APB+18] Y. Aytar, T. Pfaff, D. Budden, T. Le Paine, Z. Wang, and N. de Freitas. Playing hard exploration games by watching YouTube. *CoRR*, abs/1805.11592, 2018.

[AWR+17] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NIPS*, 2017.

[BNVB13] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade Learning Environment: An evaluation platform for general agents. *JAIR*, 47, 2013.

[BSO+16] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *NIPS 2016*, pages 1471–1479, 2016.

[CL17] Devendra Singh Chaplot and Guillaume Lample. Arnold: An autonomous agent to play FPS games. In *AAAI 2017*, pages 5085–5086, 2017.

[CLSS16] Devendra Singh Chaplot, Guillaume Lample, Kanthashree Mysore Sathyendra, and Ruslan Salakhutdinov. Transfer deep reinforcement learning in 3D environments: An empirical study. In *NIPS Deep RL Workshop*, 2016.

[DHK+17] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. OpenAI baselines. https://github.com/openai/baselines, 2017.

[DK16] Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. *CoRR*, abs/1611.01779, 2016.

[DRC+17] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: an open urban driving simulator. In *CoRL 2017*, pages 1–16, 2017.

[DSC+16] Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016.

[HVP+] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John

Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. In *AAAI 2018*.

[JLK17] Peter H. Jin, Sergey Levine, and Kurt Keutzer. Regret minimization for partially observable deep reinforcement learning. *CoRR*, abs/1710.11424, 2017.

[KSGG16] Tejas D. Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J. Gershman. Deep successor reinforcement learning. *CoRR*, abs/1606.02396, 2016.

[KSS17] Russell Kaplan, Christopher Sauer, and Alexander Sosa. Beating Atari with natural language guided reinforcement learning. *CoRR*, abs/1704.05539, 2017.

[KWR$^+$] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *CIG 2016*.

[LJA$^+$18] Hoang Minh Le, Nan Jiang, Alekh Agarwal, Miroslav Dudík, Yisong Yue, and Hal Daumé III. Hierarchical imitation and reinforcement learning. In *ICML 2018*, pages 2923–2932, 2018.

[LOB] Aravind S. Lakshminarayanan, Sherjil Ozair, and Yoshua Bengio. Reinforcement learning with few expert demonstrations. In *NIPS 2016 Workshops: Deep Learning for Action and Interaction*.

[LPCS] Lisa Lee, Emilio Parisotto, Devendra Singh Chaplot, and Ruslan Salakhutdinov. LSTM iteration networks: An exploration of differentiable path finding. In *ICLR 2018 Workshop*.

[MBM$^+$] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML 2016*.

[MKS$^+$15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[OBvdOM17] Georg Ostrovski, Marc G. Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. *CoRR*, abs/1703.01310, 2017.

[PAED17] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *CoRR*, abs/1705.05363, 2017.

[PPH$^+$18] T. Pohlen, B. Piot, T. Hester, M. Gheshlaghi Azar, D. Horgan, D. Budden, G. Barth-Maron, H. van Hasselt, J. Quan, M. Vecerík, M. Hessel, R. Munos, and O. Pietquin. Observe and look further: Achieving consistent performance on Atari. *CoRR*, abs/1805.11593, 2018.

[PS17] Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. *CoRR*, abs/1702.08360, 2017.

[RB] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *AISTATS 2010*.

[RGB] Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS 2011*.

[RKG$^+$17] A. Rajeswaran, V. Kumar, A. Gupta, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv:1709.10087*, 2017.

[SBU$^+$17] Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve J. Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *CoRR*, abs/1707.00130, 2017.

[SC18] T. Salimans and R. Chen. Learning montezuma's revenge from a single demonstration, OpenAI Blog, 2018.

[SDK18] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *CoRR*, abs/1803.00653, 2018.

[SLA$^+$15] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML, 2015*, pages 1889–1897, 2015.

[SML$^+$15] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015.

[SMSM99] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063, 1999.

[VHS$^+$] M. Večerík, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv:1707.08817*.

[VOS$^+$] A. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. *arXiv:1703.01161*.

[Waw09] Pawel Wawrzynski. Real-time reinforcement learning by sequential actor-critics and experience replay. *Neural Networks*, 22(10):1484–1497, 2009.

[WBH$^+$] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Rémi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. In *ICLR 2017*.

[Wil92] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[WML$^+$17] Yuhuai Wu, Elman Mansimov, Shun Liao, Roger B. Grosse, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *CoRR*, abs/1708.05144, 2017.

[WT] Yuxin Wu and Yuandong Tian. Training agent for first-person shooter game with actor-critic curriculum learning. In *ICLR 2017*.

[WT13] Paweł Wawrzyński and Ajay Kumar Tanwani. Autonomous reinforcement learning with experience replay. *Neural Networks*, 41:156 – 167, 2013.

[ZM18] Xiaoqin Zhang and Huimin Ma. Pretraining deep actor-critic reinforcement learning algorithms with expert demonstrations. *CoRR*, abs/1801.10459, 2018.