Supplementary Material

A. Analysis of V-trace

A.1. V-trace operator

Define the V-trace operator $\mathcal{R}$:

$$\mathcal{R}V(x) \quad \overset{\text{def}}{=} \quad V(x) + \mathbb{E}_\mu\Big[\sum_{t\geq 0}\gamma^t\big(c_0\dots c_{t-1}\big)\rho_t\big(r_t+\gamma V(x_{t+1})-V(x_t)\big)\big|x_0=x,\mu\Big], \tag{1}$$

where the expectation $\mathbb{E}_\mu$ is with respect to the policy μ which has generated the trajectory $(x_t)_{t\geq 0}$, i.e., $x_0 = x$, $x_{t+1}\sim p(\cdot|x_t,a_t)$, $a_t\sim\mu(\cdot|x_t)$. Here we consider the infinite-horizon operator but very similar results hold for the n-step truncated operator.

Theorem 1. *Let $\rho_t = \min\big(\bar\rho, \frac{\pi(a_t|x_t)}{\mu(a_t|x_t)}\big)$ and $c_t = \min\big(\bar c, \frac{\pi(a_t|x_t)}{\mu(a_t|x_t)}\big)$ be truncated importance sampling weights, with $\bar\rho\geq\bar c$. Assume that there exists $\beta\in(0,1]$ such that $\mathbb{E}_\mu\rho_0\geq\beta$. Then the operator $\mathcal{R}$ defined by (1) has a unique fixed point $V^{\pi_{\bar\rho}}$, which is the value function of the policy $\pi_{\bar\rho}$ defined by*

$$\pi_{\bar\rho}(a|x)\overset{\text{def}}{=}\frac{\min\big(\bar\rho\mu(a|x),\pi(a|x)\big)}{\sum_{b\in A}\min\big(\bar\rho\mu(b|x),\pi(b|x)\big)}, \tag{2}$$

Furthermore, $\mathcal{R}$ is a η-contraction mapping in sup-norm, with

$$\eta\overset{\text{def}}{=}\gamma^{-1}-(\gamma^{-1}-1)\mathbb{E}_\mu\big[\sum_{t\geq 0}\gamma^t\big(\prod_{i=0}^{t-2}c_i\big)\rho_{t-1}\big]\leq 1-(1-\gamma)\beta<1.$$

Remark 1. *The truncation levels $\bar c$ and $\bar\rho$ play different roles in this operator:*

- *$\bar\rho$ impacts the fixed-point of the operator, thus the policy $\pi_{\bar\rho}$ which is evaluated. For $\bar\rho=\infty$ (untruncated ρ_t) we get the value function of the target policy V^π, whereas for finite $\bar\rho$, we evaluate a policy which is in between μ and π (and when ρ is close to 0, then we evaluate V^μ). So the larger $\bar\rho$ the smaller the bias in off-policy learning. The variance naturally grows with $\bar\rho$. However notice that we do not take the product of those ρ_t coefficients (in contrast to the c_s coefficients) so the variance does not explode with the time horizon.*
- *$\bar c$ impacts the contraction modulus η of $\mathcal{R}$ (thus the speed at which an online-algorithm like V-trace will converge to its fixed point $V^{\pi_{\bar\rho}}$). In terms of variance reduction, here is it really important to truncate the importance sampling ratios in c_t because we take the product of those. Fortunately, our result says that for any level of truncation $\bar c$, the fixed point (the value function $V^{\pi_{\bar\rho}}$ we converge to) is the same: it does not depend on $\bar c$ but on $\bar\rho$ only.*

Proof. First notice that we can rewrite $\mathcal{R}$ as

$$\mathcal{R}V(x)=(1-\mathbb{E}_\mu\rho_0)V(x)+\mathbb{E}_\mu\left[\sum_{t\geq 0}\gamma^t\Big(\prod_{s=0}^{t-1}c_s\Big)\Big(\rho_t r_t+\gamma[\rho_t-c_t\rho_{t+1}]V(x_{t+1})\Big)\right].$$

Thus

$$\mathcal{R}V_1(x)-\mathcal{R}V_2(x)=(1-\mathbb{E}_\mu\rho_0)\big[V_1(x)-V_2(x)\big]+\mathbb{E}_\mu\left[\sum_{t\geq 0}\gamma^{t+1}\Big(\prod_{s=0}^{t-1}c_s\Big)[\rho_t-c_t\rho_{t+1}]\big[V_1(x_{t+1})-V_2(x_{t+1})\big]\right].$$

$$=\mathbb{E}_\mu\left[\sum_{t\geq 0}\gamma^t\Big(\prod_{s=0}^{t-2}c_s\Big)\underbrace{[\rho_{t-1}-c_{t-1}\rho_t]}_{\alpha_t}\big[V_1(x_t)-V_2(x_t)\big]\right],$$

with the notation that $c_{-1} = \rho_{-1} = 1$ and $\prod_{s=0}^{t-2} c_s = 1$ for $t = 0$ and 1. Now the coefficients $(\alpha_t)_{t\geq 0}$ are non-negative in expectation. Indeed, since $\bar{\rho} \geq \bar{c}$, we have

$$\mathbb{E}_\mu \alpha_t = \mathbb{E}\big[\rho_{t-1} - c_{t-1}\rho_t\big] \geq \mathbb{E}_\mu\big[c_{t-1}(1-\rho_t)\big] \geq 0,$$

since $\mathbb{E}_\mu \rho_t \leq \mathbb{E}_\mu\big[\frac{\pi(a_t|x_t)}{\mu(a_t|x_t)}\big] = 1$. Thus $V_1(x) - V_2(x)$ is a linear combination of the values $V_1 - V_2$ at other states, weighted by non-negative coefficients whose sum is

$$\begin{aligned}
& \sum_{t\geq 0} \gamma^t \mathbb{E}_\mu \left[\Big(\prod_{s=0}^{t-2} c_s \Big) [\rho_{t-1} - c_{t-1}\rho_t] \right] \\
= \quad & \sum_{t\geq 0} \gamma^t \mathbb{E}_\mu \left[\Big(\prod_{s=0}^{t-2} c_s \Big) \rho_{t-1} \right] - \sum_{t\geq 0} \gamma^t \mathbb{E}_\mu \left[\Big(\prod_{s=0}^{t-1} c_s \Big) \rho_t \right] \\
= \quad & \sum_{t\geq 0} \gamma^t \mathbb{E}_\mu \left[\Big(\prod_{s=0}^{t-2} c_s \Big) \rho_{t-1} \right] - \gamma^{-1} \left(\sum_{t\geq 0} \gamma^t \mathbb{E}_\mu \left[\Big(\prod_{s=0}^{t-2} c_s \Big) \rho_{t-1} \right] - 1 \right) \\
= \quad & \gamma^{-1} - (\gamma^{-1} - 1) \underbrace{\sum_{t\geq 0} \gamma^t \mathbb{E}_\mu \left[\Big(\prod_{s=0}^{t-2} c_s \Big) \rho_{t-1} \right]}_{\geq 1 + \gamma \mathbb{E}_\mu \rho_0} \\
\leq \quad & 1 - (1-\gamma)\mathbb{E}_\mu \rho_0 \\
\leq \quad & 1 - (1-\gamma)\beta \\
< \quad & 1.
\end{aligned}$$

We deduce that $\|\mathcal{R}V_1(x) - \mathcal{R}V_2(x)\| \leq \eta \|V_1 - V_2\|_\infty$, with $\eta = \gamma^{-1} - (\gamma^{-1} - 1) \sum_{t\geq 0} \gamma^t \mathbb{E}_\mu \Big[\Big(\prod_{s=0}^{t-2} c_s \Big) \rho_{t-1} \Big] \leq 1 - (1-\gamma)\beta < 1$, so $\mathcal{R}$ is a contraction mapping. Thus $\mathcal{R}$ possesses a unique fixed point. Let us now prove that this fixed point is $V^{\pi_{\bar{\rho}}}$. We have:

$$\begin{aligned}
& \mathbb{E}_\mu \big[\rho_t \big(r_t + \gamma V^{\pi_{\bar{\rho}}}(x_{t+1}) - V^{\pi_{\bar{\rho}}}(x_t) \big) \big| x_t \big] \\
= \quad & \sum_a \mu(a|x_t) \min \Big(\bar{\rho}, \frac{\pi(a|x_t)}{\mu(a|x_t)} \Big) \Big[r(x_t, a) + \gamma \sum_y p(y|x_t, a) V^{\pi_{\bar{\rho}}}(y) - V^{\pi_{\bar{\rho}}}(x_t) \Big] \\
= \quad & \underbrace{\sum_a \pi_{\bar{\rho}}(a|x_t) \Big[r(x_t, a) + \gamma \sum_y p(y|x_t, a) V^{\pi_{\bar{\rho}}}(y) - V^{\pi_{\bar{\rho}}}(x_t) \Big]}_{=0} \sum_b \min \big(\bar{\rho}\mu(b|x_t), \pi(b|x_t) \big) \\
= \quad & 0,
\end{aligned}$$

since this is the Bellman equation for $V^{\pi_{\bar{\rho}}}$. We deduce that $\mathcal{R}V^{\pi_{\bar{\rho}}} = V^{\pi_{\bar{\rho}}}$, thus $V^{\pi_{\bar{\rho}}}$ is the unique fixed point of $\mathcal{R}$. □

A.2. Online learning

Theorem 2. *Assume a tabular representation, i.e. the state and action spaces are finite. Consider a set of trajectories, with the k^{th} trajectory $x_0, a_0, r_0, x_1, a_1, r_1, \ldots$ generated by following μ: $a_t \sim \mu(\cdot|x_t)$. For each state x_s along this trajectory, update*

$$V_{k+1}(x_s) = V_k(x_s) + \alpha_k(x_s) \sum_{t\geq s} \gamma^{t-s} \big(c_s \ldots c_{t-1} \big) \rho_t \big(r_t + \gamma V_k(x_{t+1}) - V_k(x_t) \big), \tag{3}$$

with $c_i = \min\big(\bar{c}, \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)}\big)$, $\rho_i = \min\big(\bar{\rho}, \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)}\big)$, $\bar{\rho} \geq \bar{c}$. Assume that (1) all states are visited infinitely often, and (2) the stepsizes obey the usual Robbins-Munro conditions: for each state x, $\sum_k \alpha_k(x) = \infty$, $\sum_k \alpha_k^2(x) < \infty$. Then $V_k \to V^{\pi_{\bar{\rho}}}$ almost surely.

The proof is a straightforward application of the convergence result for stochastic approximation algorithms to the fixed point of a contraction operator, see e.g. Dayan & Sejnowski (1994); Bertsekas & Tsitsiklis (1996); Kushner & Yin (2003).

A.3. On the choice of q_s in policy gradient

The policy gradient update rule (4) makes use of the coefficient $q_s = r_s + \gamma v_{s+1}$ as an estimate of $Q^{\pi_{\bar\rho}}(x_s, a_s)$ built from the V-trace estimate v_{s+1} at the next state x_{s+1}. The reason why we use q_s instead of v_s as target for our Q-value $Q^{\pi_{\bar\rho}}(x_s, a_s)$ is to make sure our estimate of the Q-value is as unbiased as possible, and the first requirement is that it is entirely unbiased in the case of perfect representation of the V-values. Indeed, assuming our value function is correctly estimated at all states, i.e. $V = V^{\pi_{\bar\rho}}$, then we have $\mathbb{E}[q_s|x_s, a_s] = Q^{\pi_{\bar\rho}}(x_s, a_s)$ (whereas we do not have this property for v_t). Indeed,

$$\begin{aligned}
\mathbb{E}[q_s|x_s, a_s] &= r_s + \gamma \mathbb{E}\big[V^{\pi_{\bar\rho}}(x_{s+1}) + \delta_{s+1} V^{\pi_{\bar\rho}} + \gamma c_{s+1} \delta_{s+2} V^{\pi_{\bar\rho}} + \dots\big] \\
&= r_s + \gamma \mathbb{E}\big[V^{\pi_{\bar\rho}}(x_{s+1})\big] \\
&= Q^{\pi_{\bar\rho}}(x_s, a_s)
\end{aligned}$$

whereas

$$\begin{aligned}
\mathbb{E}[v_s|x_s, a_s] &= V^{\pi_{\bar\rho}}(x_s) + \rho_s\big(r_s + \gamma \mathbb{E}\big[V^{\pi_{\bar\rho}}(x_{s+1})\big] - V^{\pi_{\bar\rho}}(x_s)\big) + \gamma c_s \delta_{s+1} V^{\pi_{\bar\rho}} + \dots \\
&= V^{\pi_{\bar\rho}}(x_s) + \rho_s\big(r_s + \gamma \mathbb{E}\big[V^{\pi_{\bar\rho}}(x_{s+1})\big] - V^{\pi_{\bar\rho}}(x_s)\big) \\
&= V^{\pi_{\bar\rho}}(x_s)(1 - \rho_s) + \rho_s Q^{\pi_{\bar\rho}}(x_s, a_s),
\end{aligned}$$

which is different from $Q^{\pi_{\bar\rho}}(x_s, a_s)$ when $V^{\pi_{\bar\rho}}(x_s) \neq Q^{\pi_{\bar\rho}}(x_s, a_s)$.

B. Reference Scores

Task t	**Human** h	**Random** r	**Experts**	**IMPALA**
rooms_collect_good_objects_test	10.0	0.1	9.0	5.8
rooms_exploit_deferred_effects_test	85.7	8.5	15.6	11.0
rooms_select_nonmatching_object	65.9	0.3	7.3	26.1
rooms_watermaze	54.0	4.1	26.9	31.1
rooms_keys_doors_puzzle	53.8	4.1	28.0	24.3
language_select_described_object	389.5	-0.1	324.6	593.1
language_select_located_object	280.7	1.9	189.0	301.7
language_execute_random_task	254.1	-5.9	-49.9	66.8
language_answer_quantitative_question	184.5	-0.3	219.4	264.0
lasertag_one_opponent_large	12.7	-0.2	-0.2	0.3
lasertag_three_oponents_large	18.6	-0.2	-0.1	4.1
lasertag_one_opponent_small	18.6	-0.1	-0.1	2.5
lasertag_three_opponents_small	31.5	-0.1	19.1	11.3
natlab_fixed_large_map	36.9	2.2	34.7	12.2
natlab_varying_map_regrowth	24.4	3.0	20.7	15.9
natlab_varying_map_randomized	42.4	7.3	36.1	29.0
skymaze_irreversible_path_hard	100.0	0.1	13.6	30.0
skymaze_irreversible_path_varied	100.0	14.4	45.1	53.6
pyschlab_arbitrary_visuomotor_mapping	58.8	0.2	16.4	14.3
pyschlab_continuous_recognition	58.3	0.2	29.9	29.9
pyschlab_sequential_comparison	39.5	0.1	0.0	0.0
pyschlab_visual_search	78.5	0.1	0.0	0.0
explore_object_locations_small	74.5	3.6	57.8	62.6
explore_object_locations_large	65.7	4.7	37.0	51.1
explore_obstructed_goals_small	206.0	6.8	135.2	188.8
explore_obstructed_goals_large	119.5	2.6	39.5	71.0
explore_goal_locations_small	267.5	7.7	209.4	252.5
explore_goal_locations_large	194.5	3.1	83.1	125.3
explore_object_rewards_few	77.7	2.1	39.8	43.2
explore_object_rewards_many	106.7	2.4	58.7	62.6
Mean Capped Normalised Score: $\left(\sum_t \min\left[1, (s_t - r_t)/(h_t - r_t)\right]\right)/N$	100%	0%	44.5%	49.4%

Table B.1. DMLab-30 test scores.

B.1. Final training scores on DMLab-30

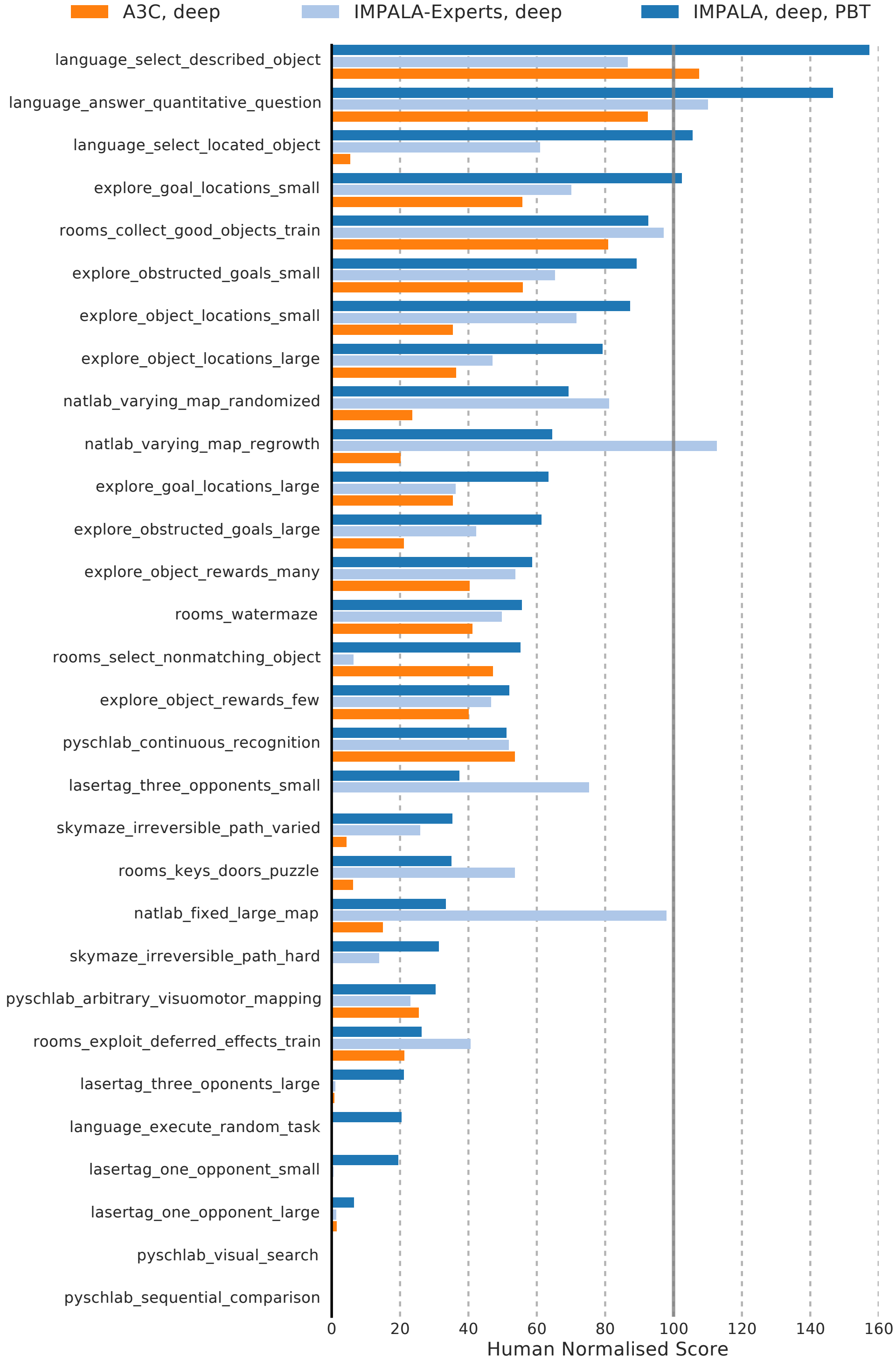

Figure B.1. Human normalised scores across all DMLab-30 tasks.

C. Atari Scores

	ACKTR	The Reactor	IMPALA (deep, multi-task)	IMPALA (shallow)	IMPALA (deep)
alien	3197.10	6482.10	2344.60	1536.05	**15962.10**
amidar	1059.40	833	136.82	497.62	**1554.79**
assault	10777.70	11013.50	2116.32	12086.86	**19148.47**
asterix	31583.00	36238.50	2609.00	29692.50	**300732.00**
asteroids	34171.60	2780.40	2011.05	3508.10	**108590.05**
atlantis	**3433182.00**	308258	460430.50	773355.50	849967.50
bank_heist	**1289.70**	988.70	55.15	1200.35	1223.15
battle_zone	8910.00	**61220**	7705.00	13015.00	20885.00
beam_rider	13581.40	8566.50	698.36	8219.92	**32463.47**
berzerk	927.20	1641.40	647.80	888.30	**1852.70**
bowling	24.30	**75.40**	31.06	35.73	59.92
boxing	1.45	99.40	96.63	96.30	**99.96**
breakout	735.70	518.40	35.67	640.43	**787.34**
centipede	7125.28	3402.80	4916.84	5528.13	**11049.75**
chopper_command	N/A	**37568**	5036.00	5012.00	28255.00
crazy_climber	150444.00	**194347**	115384.00	136211.50	136950.00
defender	N/A	113128	16667.50	58718.25	**185203.00**
demon_attack	**274176.70**	100189	10095.20	107264.73	132826.98
double_dunk	-0.54	**11.40**	-1.92	-0.35	-0.33
enduro	0.00	**2230.10**	971.28	0.00	0.00
fishing_derby	33.73	23.20	35.27	32.08	**44.85**
freeway	0.00	**31.40**	21.41	0.00	0.00
frostbite	N/A	**8042.10**	2744.15	269.65	317.75
gopher	47730.80	**69135.10**	913.50	1002.40	66782.30
gravitar	N/A	**1073.80**	282.50	211.50	359.50
hero	N/A	**35542.20**	18818.90	33853.15	33730.55
ice_hockey	-4.20	3.40	-13.55	-5.25	**3.48**
jamesbond	490.00	**7869.20**	284.00	440.00	601.50
kangaroo	3150.00	**10484.50**	8240.50	47.00	1632.00
krull	9686.90	9930.80	**10807.80**	9247.60	8147.40
kung_fu_master	34954.00	**59799.50**	41905.00	42259.00	43375.50
montezuma_revenge	N/A	**2643.50**	0.00	0.00	0.00
ms_pacman	N/A	2724.30	3415.05	6501.71	**7342.32**
name_this_game	N/A	9907.20	5719.30	6049.55	**21537.20**
phoenix	133433.70	40092.20	7486.50	33068.15	**210996.45**
pitfall	**-1.10**	-3.50	-1.22	-11.14	-1.66
pong	20.90	20.70	8.58	20.40	**20.98**
private_eye	N/A	**15177.10**	0.00	92.42	98.50
qbert	23151.50	22956.50	10717.38	18901.25	**351200.12**
riverraid	17762.80	16608.30	2850.15	17401.90	**29608.05**
road_runner	53446.00	**71168**	24435.50	37505.00	57121.00
robotank	16.50	**68.50**	9.94	2.30	12.96
seaquest	1776.00	**8425.80**	844.60	1716.90	1753.20
skiing	N/A	-10753.40	**-8988.00**	-29975.00	-10180.38
solaris	2368.60	**2760**	1160.40	2368.40	2365.00
space_invaders	19723.00	2448.60	199.65	1726.28	**43595.78**
star_gunner	82920.00	70038	1855.50	69139.00	**200625.00**
surround	N/A	6.70	-8.51	-8.13	**7.56**
tennis	N/A	**23.30**	-8.12	-1.89	0.55
time_pilot	22286.00	19401	3747.50	6617.50	**48481.50**
tutankham	**314.30**	272.60	105.22	267.82	292.11
up_n_down	**436665.80**	64354.20	82155.30	273058.10	332546.75
venture	N/A	**1597.50**	1.00	0.00	0.00
video_pinball	100496.60	469366	20125.14	228642.52	**572898.27**
wizard_of_wor	702.00	**13170.50**	2106.00	4203.00	9157.50
yars_revenge	**125169.00**	102760	14739.41	80530.13	84231.14
zaxxon	17448.00	25215.50	6497.00	1148.50	**32935.50**

Table C.1. Atari scores after 200M steps environment steps of training. Up to 30 no-ops at the beginning of each episode.

D. Parameters

In this section, the specific parameter settings that are used throughout our experiments are given in detail.

Hyperparameter	Range	Distribution
Entropy regularisation	[5e-5, 1e-2]	Log uniform
Learning rate	[5e-6, 5e-3]	Log uniform
RMSProp epsilon (ε) regularisation parameter	[1e-1, 1e-3, 1e-5, 1e-7]	Categorical

Table D.1. The ranges used in sampling hyperparameters across all experiments that used a sweep and for the initial hyperparameters for PBT. Sweep size and population size are 24. Note, the loss is *summed* across the batch and time dimensions.

Action	Native DeepMind Lab Action
Forward	[0, 0, 0, 1, 0, 0, 0]
Backward	[0, 0, 0, -1, 0, 0, 0]
Strafe Left	[0, 0, -1, 0, 0, 0, 0]
Strafe Right	[0, 0, 1, 0, 0, 0, 0]
Look Left	[-20, 0, 0, 0, 0, 0, 0]
Look Right	[20, 0, 0, 0, 0, 0, 0]
Forward + Look Left	[-20, 0, 0, 1, 0, 0, 0]
Forward + Look Right	[20, 0, 0, 1, 0, 0, 0]
Fire	[0, 0, 0, 0, 1, 0, 0]

Table D.2. Action set used in all tasks from the DeepMind Lab environment, including the DMLab-30 experiments.

D.1. Fixed Model Hyperparameters

In this section, we list all the hyperparameters that were kept fixed across all experiments in the paper which are mostly concerned with observations specifications and optimisation. We first show below the reward pre-processing function that is used across all experiments using DeepMind Lab, followed by all fixed numerical values.

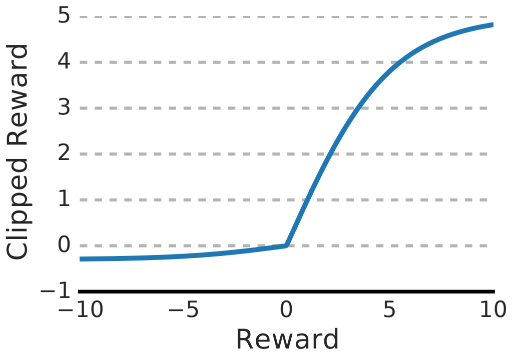

Figure D.1. Optimistic Asymmetric Clipping - $0.3 \cdot \min(\tanh(reward), 0) + 5.0 \cdot \max(\tanh(reward), 0)$

Parameter	**Value**
Image Width	96
Image Height	72
Action Repetitions	4
Unroll Length (n)	100
Reward Clipping	
- Single tasks	[-1, 1]
- DMLab-30, including experts	See Figure D.1
Discount (γ)	0.99
Baseline loss scaling	0.5
RMSProp momentum	0.0
Experience Replay (in Section 5.2.2)	
- Capacity	10,000 trajectories
- Sampling	Uniform
- Removal	First-in-first-out

Table D.3. Fixed model hyperparameters across all DeepMind Lab experiments.

E. V-trace Analysis

E.1. Controlled Updates

Here we show how different algorithms (On-Policy, No-correction, ε-correction, V-trace) behave under varying levels of policy-lag between the actors and the learner.

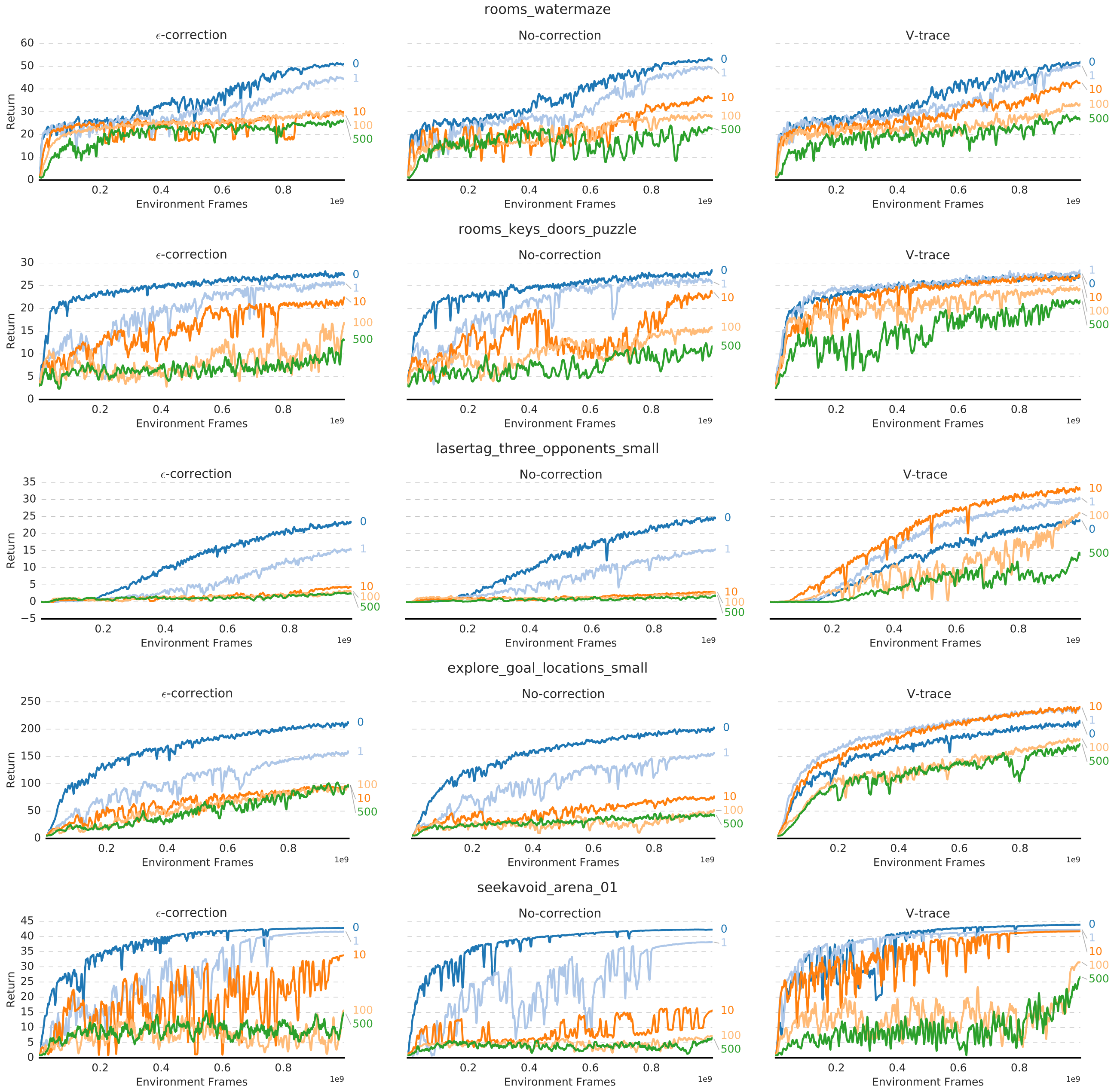

Figure E.1. As the policy-lag (the number of update steps the actor policy is behind learner policy) increases, learning with V-trace is more robust compared to ε-correction and pure on-policy learning.

E.2. V-trace Stability Analysis

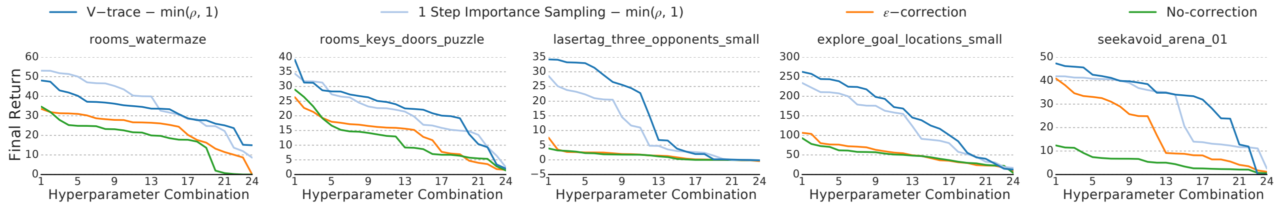

Figure E.2. Stability across hyper parameter combinations for different off-policy correction variants using replay. V-trace is much more stable across a wide range of parameter combinations compared to ε-correction and pure on-policy learning.

E.3. Estimating the State Action Value for Policy Gradient

We investigated different ways of estimating the state action value function used to estimate advantages for the policy gradient calculation. The variant presented in the main section of the paper uses the V-trace corrected value function v_{s+1} to estimate $q_s = r_s + \gamma v_{s+1}$. Another possibility is to use the actor-critic baseline $V(x_{s+1})$ to estimate $q_s = r_s + \gamma V(x_{s+1})$. Note that the latter variant does not use any information from the current policy rollout to estimate the policy gradient and relies on an accurate estimate of the value function. We found the latter variant to perform worse both when comparing the top 3 runs and an average over all runs of the hyperparameter sweep as can be see in figures E.3 and E.4.

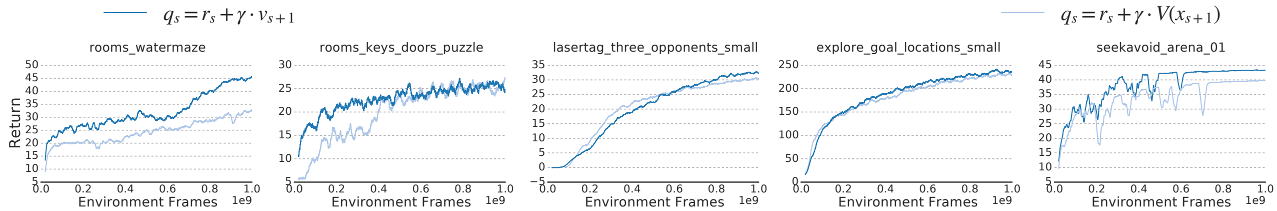

Figure E.3. Variants for estimation of state action value function - average over top 3 runs.

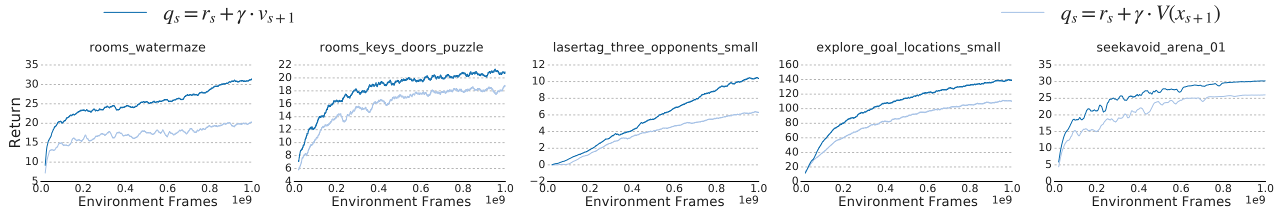

Figure E.4. Variants for estimation of state action value function - average over all runs.

F. Population Based Training

For Population Based Training we used a “burn-in” period of 20 million frames where no evolution is done. This is to stabilise the process and to avoid very rapid initial adaptation which hinders diversity. After collecting 5,000 episode rewards in total, the mean capped human normalised score is calculated and a random instance in the population is selected. If the score of the selected instance is more than an absolute 5% higher, then the selected instance weights and parameters are copied.

No matter if a copy happened or not, each parameter (RMSProp epsilon, learning rate and entropy cost) is permuted with 33% probability by multiplying with either 1.2 or $1/1.2$. This is different from Jaderberg et al. (2017) in that our multiplication is unbiased where they use a multiplication of 1.2 or $.8$. We found that diversity is increased when the parameters are permuted even if no copy happened.

We reconstruct the learning curves of the PBT runs in Figure 5 by backtracking through the ancestry of copied checkpoints for selected instances.

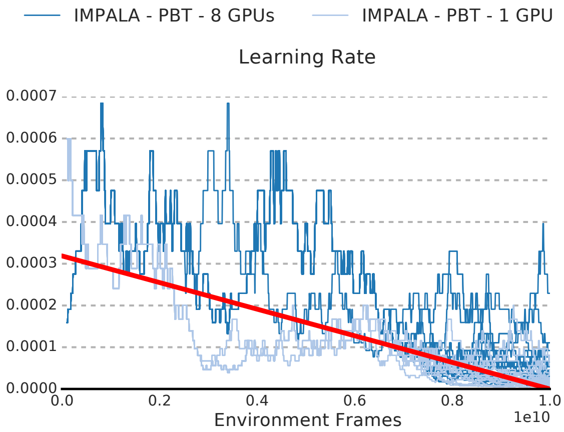

Figure F.1. Learning rate schedule that is discovered by the PBT Jaderberg et al. (2017) method compared against the linear annealing schedule of the best run from the parameter sweep (red line).

G. Atari Experiments

All agents trained on Atari are equipped only with a feed forward network and pre-process frames in the same way as described in Mnih et al. (2016). When training experts agents, we use the same hyperparameters for each game for both IMPALA and A3C. These hyperparameters are the result of tuning A3C with a shallow network on the following games: `breakout`, `pong`, `space_invaders`, `seaquest`, `beam_rider`, `qbert`. Following related work, experts use game-specific action sets.

The multi-task agent was equipped with a feed forward residual network (see Figure 3). The learning rate, entropy regularisation, RMSProp ε and gradient clipping threshold were adapted through population based training. To be able to use the same policy layer on all Atari games in the multi-task setting we train the multi-task agent on the full Atari action set consisting of 18 actions.

Agents were trained using the following set of hyperparameters:

Parameter	Value
Image Width	84
Image Height	84
Grayscaling	Yes
Action Repetitions	4
Max-pool over last N action repeat frames	2
Frame Stacking	4
End of episode when life lost	Yes
Reward Clipping	[-1, 1]
Unroll Length (n)	20
Batch size	32
Discount (γ)	0.99
Baseline loss scaling	0.5
Entropy Regularizer	0.01
RMSProp momentum	0.0
RMSProp ε	0.01
Learning rate	0.0006
Clip global gradient norm	40.0
Learning rate schedule	Anneal linearly to 0 From beginning to end of training.
Population based training (only multi-task agent)	
- Population size	24
- Start parameters	Same as DMLab-30 sweep
- Fitness	Mean capped human normalised scores $\left(\sum_l \min\left[1, (s_t - r_t)/(h_t - r_t)\right]\right)/N$
- Adapted parameters	Gradient clipping threshold Entropy regularisation Learning rate RMSProp ε

Table G.1. Hyperparameters for Atari experiments.

References

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Dayan, P. and Sejnowski, T. J. TD(λ) converges with probability 1. *Machine Learning*, 14(1):295–301, 1994. doi: 10.1023/A:1022657612745.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., and Kavukcuoglu, K. Population based training of neural networks. *CoRR*, abs/1711.09846, 2017.

Kushner, H. and Yin, G. *Stochastic Approximation and Recursive Algorithms and Applications*. Stochastic Modelling and Applied Probability. Springer New York, 2003. ISBN 9780387008943.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. *ICML*, 2016.